ABSTRACT OF THESIS

IMPLEMENTATION OF A UNIVERSAL MICRO-SENSOR INTERFACE CHIP

This thesis presents the design and implementation of an interface circuit which provides the critical link between a microcontroller and a network of sensors in low-power microsystems. The chip implements an 11-line serial communication sensor bus optimized from an IEEE standard to include plug-n-play features and support multiplexed analog output signals. The interface circuit also includes 24 8b SRAM-style data registers, a 4-node SPI, an 8b I/O port, a 5b interrupt source management unit, a mixed-signal output signal multiplexer, an on-chip temperature sensor, and a variety of highly-programmable and self-testing sensor readout circuits. Significant features of this chip are low-power design including chip-level power management, single chip solution containing all necessary interface electronics and capable of network implementation, and the ability to interface with a very wide variety of capacitive, resistive, and potentiometric sensors and actuators. The design of this chip and its implementation in a 0.5µm CMOS process are presented.

KEYWORDS: Sensor Bus Interface, Smart Sensor, Microsystem

_____

_____

IMPLEMENTATION OF A UNIVERSAL MICRO-SENSOR INTERFACE (UMSI) CHIP

THESIS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering at the University of Kentucky

By

Kun Zhang

Lexington, Kentucky

Director: Dr. Andrew Mason, Professor of

Electrical and Computer Engineering

Lexington, Kentucky

2002

TABLE OF CONTENTS

Acknowledgments ……...iii

List of Tables ……...…….vii

List of Figures ……..……..……...…viii

Chapter One: Introduction ……..……1

Chapter Two: Sensor Bus……...…….....6

2.1 IEEE P1451 Standard…….……6

2.2 An Intramodule Multielement Microsystem Bus……..……..11

Chapter Three: Universal Micro-System Interface (UMSI) Design……..……16

3.1 Chip Design Overview…….……....16

3.1.1 IM^2 Bus Compatibility…..…….....17

3.1.2 Low power Dissipation……...…18

3.1.3 Application Adaptability…...……..…...……...19

3.1.4 Design for Future Expansion…….....19

3.1.5 Interrupt Generation and Handling……...…20

3.1.6 Triggering……..…..…….20

3.1.7 SPI Port….…….……...21

3.1.8 Retrieve Chip ID from EEPROM.……21

3.1.9 New Node Detection …...…….22

3.1.10 TEDS…….....……..…...…..……...23

3.2 IM^2 Bus Interface……..….23

3.2.1 Instruction register and decoder ……...25

3.2.2 The Address Register and Write in Data Register……..26

3.2.3 The ID register and compare it with ID set in EEPROM and IO..27

3.2.4 The signals switch between EMMSB and SPI bus……..….28

3.2.5 Data (Signal) Output Path.…..……...28

3.2.6 The Memory and IO read out register ……...30

3.2.7 IO Port…….……....30

3.2.8 Timing Signal Generation.……...31

3.2.9 Power Supplies…….……...32

3.2.10 The Final Schematic.....33

3.3 On-chip SRAM and Status-readout circuit……....34

3.4 Analog Interface Circuit……....36

3.4.1 Capacitive readout, gain stage and sample & hold stage……..38

3.4.2 Resistive readout and voltage-output readout……..……..40

3.4.3 Digital-to-Analog Converter (DAC)…….42

3.5 SRAM Assignment……..……....43

Chapter Four: Chip Physical Layout Design……...45

4.1 Layout Design Overview……...45

4.2 Standard Cell Library……..……..47

4.3 Circuit Building-Block……..51

4.4 AMI05 Pad-Frame……..……..55

4.5 Layout for Testing……..55

4.6 Floor-plan, Place and Route of the Chip……59

Chapter Five: Conclusions……..……63

REFERENCES …….66

VITA …….68

LIST OF TABLES

Table 2.1, The IM^2 bus and the signals...……12

Table 3.1, Instructions set for UMSI chip....…….25

Table 3.2, The truth table of the SRAM block...……..…….36

LIST OF FIGURES

Figure 1.1, Overview of a smart sensing system...……..……..3

Figure 2.1, : Fieldbus sensor module structure and usage...……6

Figure 2.2, Smart sensor system hierarchy...…….……..……7

Figure 2.3, Physical representation of IEEE P1451...……...…9

Figure 2.4, Timing diagram for the IM^2 sensor bus communication signals……...…..…15

Figure 3.1, Block diagram of the Universal Micro-System Interface (UMSI) chip……16

Figure 3.2, UMSI gets the ID from EEPROM……...….22

Figure 3.3, Overview of UMSI interface circuit…….24

Figure 3.4, Instruction register and decoder…….……..……26

Figure 3.5, The address register and write in data register..……...26

Figure 3.6, The ID register and compare it with ID set in EEPROM and IO…….27

Figure 3.7, ID register and compare it with ID set in EEPROM and IO……...…27

Figure 3.8, The signals switch between EMMSB and SPI bus..……28

Figure 3.9, Data (Signal) output path…….…….29

Figure 3.10, The memory and IO read out register……...30

Figure 3.11, IO port…….……...30

Figure 3.12, Timing signal generation circuit..……...31

Figure 3.13, Internal signals timing……32

Figure 3.14, Schematic of UMSI bus interface (Top Level)…..……33

Figure 3.6, Stator slot dimensions…….45

Figure 3.15, On-chip SRAM schematic……..35

Figure 3.16, Architecture of the UMSI chip analog interface…..……..37

Figure 3.17, Capacitive readout, programmable gain and sample &hold stage…….39

Figure 3.18, Resistive sensor readout……..……..40

Figure 3.19, Voltage-output readout……...41

Figure 3.20, Programmable attenuator…….41

Figure 3.21, 6-bit digital-to-analog converter…….……42

Figure 3.22, Analog ground generator……..……..43

Figure 4.1, Chip floor-plan before routing…….…….47

Figure 4.2, Functional and time simulation on D flip-flop with reset……..…..48

Figure 4.3, D flip-flop with buffer inputs and outputs……50

Figure 4.4, Different height cells placed together…..…….51

Figure 4.5, Regularity of DAC current source…..……...53

Figure 4.6, AMI 0.5 Hi-ESD Minimum Pad Frame……..…….55

Figure 4.7, Testing chain…….56

Figure 4.8, The schematic of one testing node (hook)……..….57

Figure 4.9, Pitch matching……..60

Figure 4.10, Model for Vdd and Gnd paths includes bonding wires……..…….61

Figure 5.1, Final chip……...……63

Figure 5.2, Testing Result of Reading and Writing Memory……..…….65

Chapter One

Introduction

Microelectronic systems have gone through many fundamental changes due to the rapid progress achieved in microelectronic technologies; it is feasible now to realize many control/instrumentation system components as single monolithic or hybrid modules. The introduction of the microprocessor in the early 1970s revolutionized the design and use of control/instrumentation systems by allowing system operation to be defined in software and permitting a substantial increase in signal-processing and user-interface features. Analog elements (e.g., analog-to-digital converters) have also been improved substantially to satisfy high-speed and high-accuracy requirements; however, progress in the development of sensors with which these electronic systems can interface has been slow. In sensor instruments, external physical and chemical parameters are measured and converted into an electrical format using an array of sensors. The sensed data are collected, digitized, and processed using in-module (integrated or hybrid) circuitry, and are transmitted over a digital bus to a host controller. The host controller uses this information to make appropriate decisions, and feeds control information back to the external environment through an array of actuators [1]. Such systems are increasingly needed in many applications, including automotive, health care, manufacturing, environmental monitoring, industrial processing, avionics, and defense.

In sensing systems, the sensors are a critical system element often determining the accuracy of the system. However, sensors currently represent the weakest link in the development of most emerging and next-generation instrumentation, data-acquisition, and control systems. They are often unreliable, rarely offer adequate accuracy,

sometimes cost more than the controller, and provide few, if any, fault-tolerance or fault-detection capabilities. The lack of such features results in expensive system maintenance and repair costs. For example, as far as long-term reliability and fail-safe operation are concerned, sensors have been identified as the weakest link in today's automotive control systems. The development of solid-state microsensors has helped improve some performance characteristics, particularly in terms of reproducibility, size, cost, and accuracy. However, the full potential of solid-state microsensors and microactuators has not yet been realized.

In the past few years "integrated sensors" that monolithically combine the sensor structure and at least some portion of the interface and signal-processing electronics on the same substrate have begun to emerge [1]. By merging microsensors and circuits, integrated "smart" sensors can go far beyond simple tolerance, and digital compensation provided by older designs to extend overall system accuracy, dynamic range, and reliability in ways not otherwise practical, while reducing the capital and ownership costs. Another aspect that has often been neglected in considering the advantages of integrated sensors is that a variety of new sensors, which were not practical in the past, may now be feasible. There is a wide unexplored range of possibilities for combining digital processing power with current solid-state technology to produce novel sensors and instrumentation techniques. However, the full potential of integrated sensors cannot be achieved simply by replacing conventional sensors.

The most important aspect of any sensing system is the performances achieved by the entire system rather than that of a single component, such as a sensor. Indeed, users desire a high-performance sensing system, not an isolated smart sensor. Furthermore,

many features of smart sensors are only meaningful when considered in the broader context of the overall sensing system. To accommodate the full capacity of smart sensors the structure of emerging systems must change as well. Figure 1.1 shows the system architecture of a high-performance distributed sensing system that consists of four principal parts: a host computer, which performs overall system control; a bus structure, which transmit node addresses, host commands, and data to and from the sensing nodes; a microprocessor-driven sensing node, which interfaces with its sensors and the host controller, interprets and executes commands, and provides the requested information to the host controller; and, finally, a sensor front end, which contains the sensors along with the necessary circuitry.

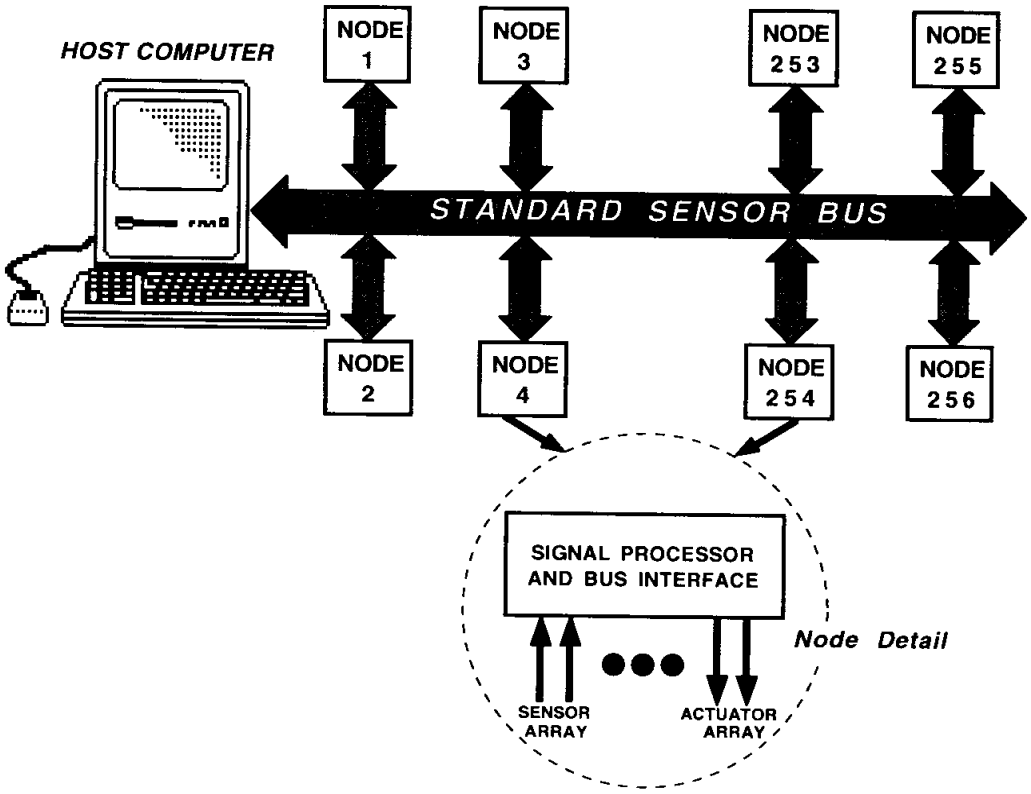


Figure 1.1: Overview of a smart sensing system [1].

Today, there has been an obvious trend toward combining the Micro-Electro-Mechanical System (MEMS) transducers with increasingly sophisticated circuits. One step in this evolution is to integrate circuitry and transducers to form "smart" sensors, yet the low cost and wide availability of present signal processing electronics make it possible to go one step further and form entire smart microsystem. It is obvious that microsystems will have a pervasive impact on the future of the microelectronics industry in application fields ranging from automotive systems to health care. The development of microsystems, miniature low-power electronics capable of interacting with the physical world, has closely paralleled the growth of smart sensors. This is largely because microsystems provide a vehicle for implementing smart sensors in more useful applications and because many microsystems rely largely on smart sensors for their interaction with the physical world. In fact, the distinction between microsystems and smart sensors is often blurred in the literature and the two terms are often used interchangeably. However, for this research, a microsystem will be defined as a system of components which may include one or more smart sensors and is therefore "greater" than an individual smart sensor. Thus, microsystems typically provide significantly more capability than a smart sensor, but generally utilize smart sensors to achieve their design goals [2].

The objective of this thesis is to define an appropriate sensor bus and develop an integrated circuit (ASIC) which will provide the necessary interface chip functions for smart sensors employed in microsystems. Design of a sensor bus requires a thorough understanding of the needs of smart sensors as well as the options and features available in existing communication buses. One goal of this research is to combine the findings in

these two areas to define a new Intramodule Multielement Microsystem (IM^2)[3] bus for the network environment of low-power, small size, microsystems. A concurrent objective is to maintain compatibility with an existing standard, if at all possible, in order to provide an improvement on a standard rather than introducing a completely new bus to a market already overwhelmed by bus options. The second major goal of this research is to create architecture for a Universal Micro-Sensor Interface (UMSI) circuit which can be universally employed in microsystems by a wide variety of transducers. This circuit should be directly compatible with the requirements of small, low-power microsystems [5]. It should also allow such systems to implement advanced features such as programmable sensor readout (gain and offset control), sensor module self test, and temperature compensation, which will support future generations of highly-functionally and highly-adaptable microsystems. In addition to defining the overall architecture for the UMSI chip, an objective of this research is to fully implement the bus interface circuitry required to utilize the IM^2 bus protocol on the interface chip.

This thesis begins with the discussion of the requirements for the sensor communication bus and the limitations in existing bus standards. Based on this information, a new sensor bus, called the Intramodule Microsystem Multielement Bus, is presented and defined. The thesis then described the architecture of Universal Micro-Sensor Interface chip along with details of the design of the interface circuitry which implements the IM^2 bus within the smart sensor module. Chip layout which includes standard cell library, building-block, whole chip placement and routing is then presented. Finally, test results from the fabricated UMSI chip are presented.

Chapter Two

Sensor Bus

2.1 IEEE P1451 Standard

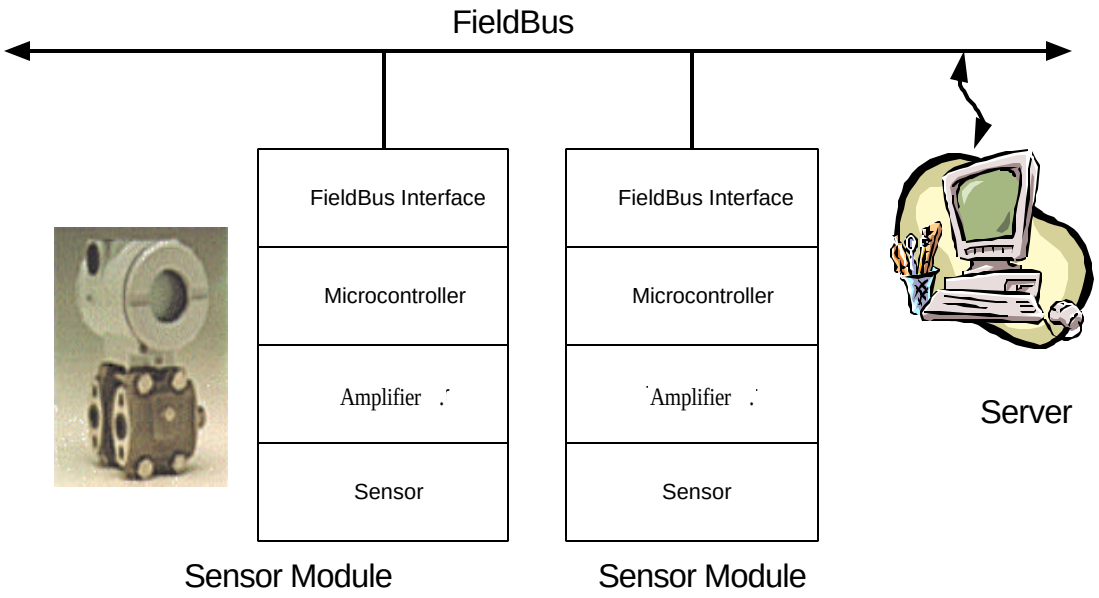


Figure 2.1: Fieldbus sensor module structure and usage[3].

Most of sensors on the market only communicate through basic point-to-point usage. This may work fine for only a few channels, but it creates a tremendous wiring overhead which can only be minimized by connecting multiple devices on a single digital bus. To build low-cost, networked smart transducers, an alternative choice is the so-called *fieldbus* [3] concept which provides a single bus with multiple nodes, shown as Figure 2.1. In a fieldbus sensor system architecture, a sensor module is a single system that includes a sensor, an amplifier, a microcontroller, and a fieldbus interface which communicates with host via fieldbus. However, fieldbus is designed for communication between microcontrollers and is not appropriate for less sophisticated sensors. Use of this bus makes it is necessary to embed a microcontroller within the sensor node to do the

sampling, compensation, and communication. In this sense the sensor node becomes not a smart sensor but a complete microcontroller-based system. As a result, a fieldbus "smart sensor" like this suffers from unnecessary complexity and size and high power consumption, and this is certainly not a good choice for microsystem applications. This approach is also limiting in that a change of a sensor module may require a large-scale change of hardware and software to match the new sensor module. There is an obvious need for a sensor bus that is better suited for microsystem applications.

A sensor bus is used to connect sensors with microprocessors. A sensor bus makes it possible to design each part of the sensor system separately. Sensor producers need not consider how the sensor will be used; they only need to provide the sensor component and relevant data (which can be saved in the EEPROM at the sensor front end). Figure 2.2 shows an example of smart sensor module hierarchy.

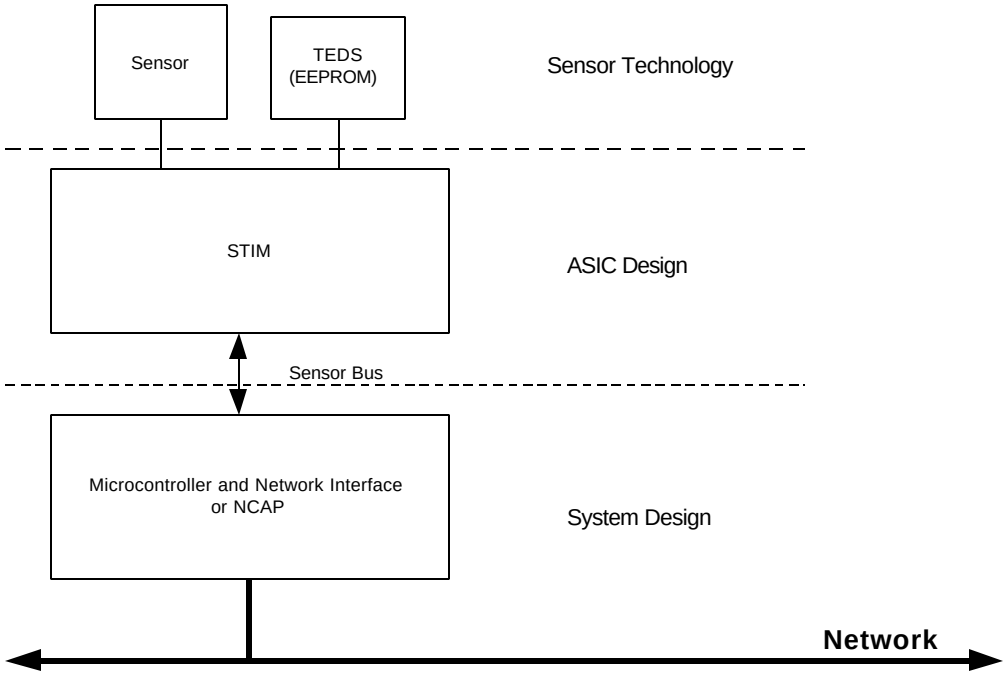


Figure 2.2: Smart sensor system hierarchy.

Here, STIM is an integrated circuit which includes the transducer interfaces for voltage, resistive and capacitive sensors. STIMs from different vendors and based on completely different underlying technology can deliver exactly the same information to any microcontroller with the necessary sensor bus hardware and software. With this architecture users will find it is very easy to build a sensor module that is compatible to any brand product, and users can remove components and plug another band into the sensor module as desired.

The IEEE P1451 Standard is a recent development that attempts to improve microsystem design by introducing a series of communication standards. It provides a smart transducer interface standard which makes it easier for transducer manufacturers to develop smart devices and to interface those devices to networks, systems, and instruments by incorporating existing and emerging sensor and networking technologies. Figure 2.3 illustrates the IEEE P1451 system with the Network-Capable Processor (NCAP) and the Smart Transducer Interface Module (STIM). The IEEE P1451.1 Standard helps to develop a standardized software framework for connecting the NCAP to control networks. It is similar to many fieldbus products in the market and defines the external interface for an IEEEP1451-based microsystem. A separate standard, IEEE P1451.2, defines the Transducer Independent Interface (TII) which connects transducers to the NCAP.

The main features of the IEEE P1451.2 standard are to:

- Enable plug and play at the transducer level by providing a common communication interface for transducers and a machine-readable TEDS.
- The control and data associated with the channel are digital.

- Triggering, status, and control are provided to support the proper functioning of the channel.

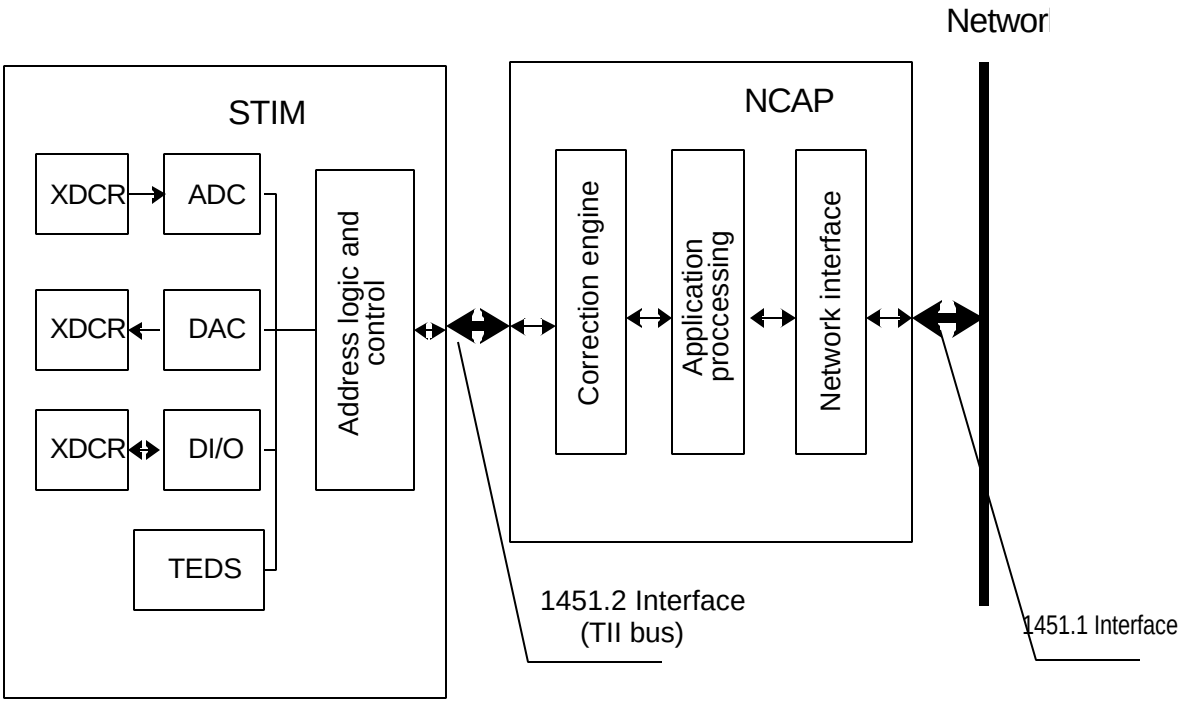


Figure 2.3: Physical representation of IEEE P1451

Figure 2.3 illustrates a typical system using the IEEE P1451.2 standard. The IEEE P1451.2 transducer, called a STIM, consists of the Transducer Electronic Data Sheet (TEDS), the control and status registers, interrupt masks, address and function-decoding logic, and trigger and trigger-acknowledge functions for the digital interface to the TII. The STIM works in conjunction with the NCAP to allow a host to collect data from a network of STIM modules. The STIM may also include application specific data processing and control functionality.

Operations that occur in the TII sensor bus are: TEDS read, in which the microcontroller in the NCAP reads the data in the STIM electronic data sheet; sensor

read, in which the NCAP accepts data from the sensor; TEDS write, in which the microcontroller writes data (for example, error-correction coefficients) to the TEDS; and actuator write, in which the NCAP sends data to an actuator. In the sensor-read operation, the NCAP triggers the STIM to begin sensor reading. Upon completion of an instruction, the STIM issues a trigger-acknowledge signal, and the NCAP reads the sensor data. In the actuator-write operation, the NCAP writes values and triggers the STIM. The STIM then performs actuation and issues a trigger-acknowledge signal. The TII bus is the heart and soul of IEEE P1451 which makes it possible for sensors to be "smart".

IEEE P1451.2 allows sensors to move into the digital age. However, for many applications it is too complex in many ways and lacks other several very useful features. Notice, with this standard, the STIM must be a complete microcontroller-based system. Under this standard, someone designing a STIM must understand how to design both a transducer and microcontroller system. This is the same problem encountered with the existing fieldbus systems. Thus, this standard does not reduce the work of system designer. In this type of system, we might as well do away with the sensor bus because the microcontroller in a STIM can also provide the functions of the NCAP and handle network communication, just like in existing fieldbus sensor modules. In most situations, two microcontrollers will be much more expensive than one, and obviously this configuration will consume more power and remain complex for the system designer. Even when faced with a hot sensor market, most manufacturers do not design using the IEEE P1451.2. The reason they do not is because it is nearly impossible to implement the standard without an embedded microcontroller which adds significant cost and development time to the sensor product. Another issue with the IEEE P1451.2 standard

is that it requires the STIM to have a repository (E/EEPROM) for the TEDS. The TEDS is comprised eight fields (two mandatory and six machine readable) each consisting of length (byte count), data and checksum bytes. Although this datasheet permits the host to find out what is on the other end of the wire with enough detail to support a wide range of applications (a sensor system version of plug-and-play), for many products the TEDS is simply too lengthy and complex requiring the added expense of an embedded microcontroller with a significant amount of memory.

To solve the problems and overcome the limitations in the sensor bus structures presented above, a new sensor bus which provides a standard link between microcontroller and sensors was developed through this thesis project and is introduced here. The new bus separates the sensor from the microcontroller and makes it possible for sensor producers to design a sensor element alone without need to match to a microcontroller. According to the new sensor bus, the STIM can be a passive ASIC where only the sensor bus clock is needed (no internal clock). This can save a lot of power when the sensor system is working in sleep mode and the sensor bus clock is disabled. In addition, the new sensor bus will support distributed multidrop sensor modules, a feature not currently available with the IEEE P1451.2 standard.

2.2 An Intramodule Multielement Microsystem Bus

The Intramodule Multielement Microsystem (IM^2) bus is an expansion of IEEE P1451.2 TII sensor bus which has been developed to correct the shortcomings of this standard. It is designed to be superset, or extension, of the TII bus and to be fully hardware compatible with the P1451.2 standard. Table 1 shows the signals of the IM^2 bus. This bus defines a set of specifications that allows a transducer manufacturer to

build transducers that have a wide range of price and performance but can all be utilized within the same system. Unlike the P1451.2 standard, the IM^2 Bus is designed to have a single bus controller (NCAP) and many sensor nodes (STIMs) supporting distributed multidrop sensor modules. The IM^2 bus also allows sensor data in both analog and digital formats and offers several power management features such as support of normally-off operation. All of these features are included to support a class of multi-sensor microsystems that feature attributes of low power dissipation and small size.

Table 2.1: The IM^2 bus and the signals

Line	**Logic**	**Description**	**Driver**
DIN	Positive	Address and data from transport from NCAP to STIMS	NCAP
DOUT	Positive	Data transport from STIM to NCAP	STIM
DCLK*	Positive	Positive-going edge latches data on DIN and DOUT	NCAP
NIOE	Active Low	Signals that data transport is active and delimits data transport framing	NCAP
NTRIG	Negative	Performs triggering function	NCAP
NACK	Negative	Serves two functions: trigger acknowledge and data transport acknowledge	STIM
NINT	Negative	Used by the STIM to request service from the NCAP	STIM
NSDET	Active Low	Used by the NCAP to detect the presence of a STIM	STIM
POWER	N/A	Normal 3-V power supply	NCAP
VSWTCH	N/A	Controllable 3-V power supply. It will be closed when system in sleep mode	
COMMON/GND	N/A	Signal common or ground	NCAP

*DCLK need not have a constant frequency or duty cycles

As shown in Table 1, the electrical connections (Sensor Bus signals) are relatively straightforward. It consists of a clocked serial interface with most of the action revolving around a data clock (DCLK) and unidirectional data lines (DIN and DOUT).

NIOE is an output enable driven by the NCAP that frames activity on the data lines. NACK is driven by the STIM to indicate that a byte transfer can proceed, i.e., that the host can continue to drive DCLK. Using NACK as a byte handshake reduces the timing burden on the STIM. The timing specification, which is shown in detail below, calls for all devices involved to support a typical 6-kHz DCLK, but it is allowed the components to mutually agree on a higher rate. NTRIG can be asserted by the host to initiate a particular operation in the STIM. This signal allows sensor establish precise timing when necessary, independent of any latency or uncertainty associated with communication and setup.

For the most part, all activities are carried out under direction of the host, except that NINT can be driven by the STIM to request service asynchronously. However, even in this case, the host response timing is not restricted. Thus, the host is completely in charge, which means that practically any device, fast or slow, can fill the role. NINT can be disabled or the interrupt resource can be reset by a command from the host.

NSDET is also driven by the STIMs as a way to signal STIMs' ID loaded and the new STIM presence. It might be pulled up on the host when no new node adds into the IM^2 Bus. NSDET will be pulled down when all or any node is powered up or re-powered up. Most time, the host may work in sleep mode and close the power of STIMs. When STIMs are powered up again, each STIM will pull down the NSDET until the host sends the command to ask them to load ID. Load ID has two cases: load ID form IO port and

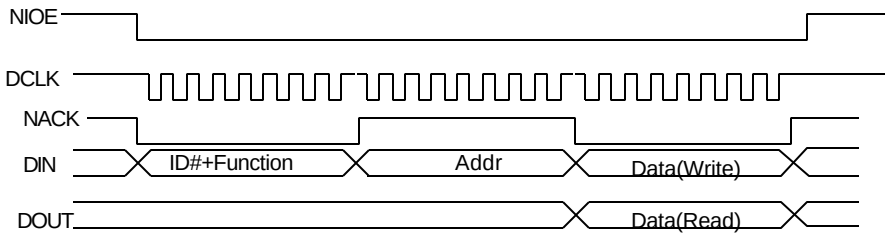
load ID from EEPROM(TEDS). At the case of load ID from IO, ID is wired bond to IO port by user. The host scans the whole possible IDs to find which is occupied. At the case of load ID from EEPROM, EEPROM of new STIM will be blank. When a new STIM is added to the bus, the NSDET will not be restored to pulling up situation until the host allocates a new ID. The new ID will be saved to EEPROM. NSDET is used as a plug-n-play indicator and help the host to manager and allocate the ID.

Although hardware consistency with the IEEE P1451.2 standard has been maintained (i.e., all of the same signals used have the same logic and function), the data protocol has been modified to fit the needs of microsystems with multiple sensor nodes, while IEEE P1451.2 protocol assumes a point-to-point connection. The primary modifications include:

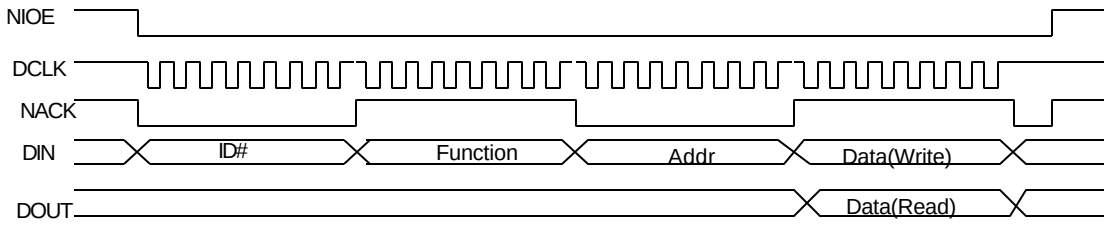
- addition of a chip address byte (ID#) at the beginning of each instruction
- implementation of several addressing modes:
    - compact address mode (chip address and function in one 8-bit word)
    - full address mode (chip address and function in individual words)
    - broadcast mode (for single-chip mode or broadcast to all nodes at the same time)
- modification of the NSDET signal function to serve as a plug-and-play indicator
- definition of DOUT as either digital, analog, or frequency-coded data
- optional addition of a the second power line to support power management features

A timing diagram of the IM^2 bus is shown in Figure 2.4. Additional details about the IM^2 bus have been described in [3] and will not be covered in this document which

will instead focus on the implementation of this bus structure and protocol in an integrated sensor interface circuit.



Compact Addr Mode --support 16 ID number



Full Addr Mode --support 255 ID number

Figure 2.4: Timing diagram for the IM^2 sensor bus communication signals.

Chapter Three

Universal Micro-System Interface (UMSI) Design

3.1 Chip Design Overview

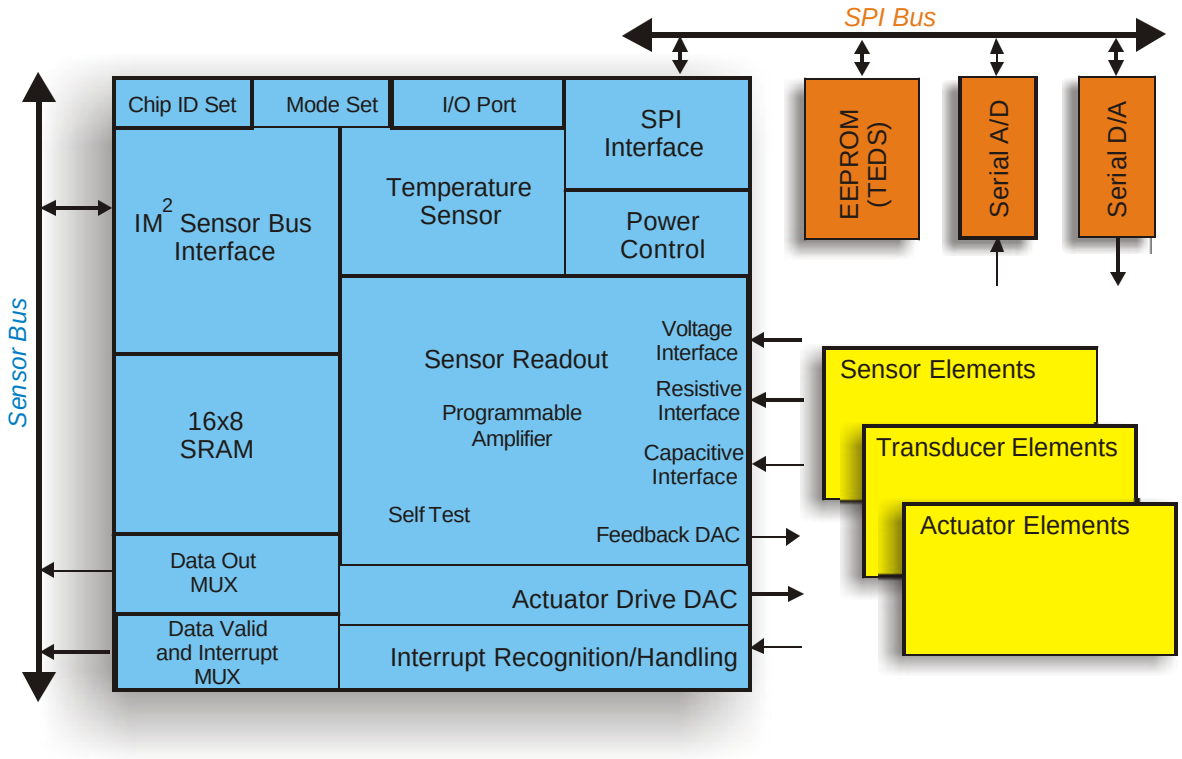


Figure 3.1: Block diagram of the Universal Micro-System Interface (UMSI) chip

A Universal Micro-System Interface (UMSI) circuit has been designed to server the needs of many different sensors which can be employed in a wide range of microsystems. Figure 3.1 shows a block diagram of this chip which is designed to provide a single chip link between various sensors and actuators and a microsystem controller. The chip implements an IM^2 bus interface unit, and includes a multipurpose analog sensor readout circuit, an on-chip integrated circuit temperature sensor, and many other features to provide a highly programmable interface between the sensor bus and a

various transducers. Communication with the microcontroller is performed over the sensor bus and is handled by the IM^2 sensor bus interface. Serial data transmitted over the bus is received, decoded and applied to control various chip blocks. Through on-chip SRAM, a microcontroller can function the UMSI chip, including transform the IM^2 bus into SPI mode; control data output multiplexer to select one out of 8 analog signals; control over all switches in sensor readout circuit; etc. The sensor readout circuit is highly programmable and designed for use by multiple sensor types. It utilizes a low-noise front-end charge integrator to read out up to 8 capacitive sensors, a separate amplifier tailored to resistive sensors, and a third input amplifier for sensor with high or low level voltage output. The sensor readout circuit also supports several methods of self-test. A temperature sensor is integrated on the interface chip to measure the temperature in close proximity to the other sensors on the microsystem. It provides both raw temperature data and can be used to digitally compensate for the temperature sensitivity of any other sensors connected through the UMSI chip. The overall chip includes the following features:

3.1.1 IM^2 Bus Compatibility

The UMSI chip provides an interface to the IM^2 sensor bus which supports the design of sensor modules which can be employed in microsystems with suite of front-end sensor nodes. Compatibility with the IM^2 bus allows the UMSI chip to be configured by the host with configuration data stored in on-chip RAM. IM^2 bus compatibility also allows the UMSI chip to support plug-and-play features provided by the bus architecture. The UMSI chip supports the Transducer Electronic Data Sheet (TEDS) feature of the bus via an external EEPROM which is accessed through the SPI interface on the chip. When

a new sensor module is added to the system, the system first reads the TEDS data. This information can be used to configure the UMSI chip for online sensor operation control such as sensor range selection, sensor readout control (gain and offset) and parameter threshold selection (shock sensor). Once configured via commands through the IM^2 bus, the system can then retrieve data from the sensor and calibrate the reading based on calibration and compensation data provided in the TEDS.

3.1.2 Low power Dissipation

UMSI chip is designed to support microsystems which operate under sever power limitations. In such systems, it is assumed that the microcontroller will normally be in low-power sleep mode and will wake to interact with the sensor front end for the following two actions:

- To perform a periodic scan of the sensors, the system will wake up to sample data after a long time sleeping. Once the system finishes this task, a new sleep cycle will begin.
- To provide for sensor-driven interrupts which will wake the system to request attention, UMSI chip utilizes the interrupt line of the IM^2 bus. In this event, the controller wakes up to check which sensor node generated the interrupt signal and then processes the interrupt.

To support these to actions, the IM^2 bus offers two power lines. One is the normal constant, "always on" power line and another is the controllable reference power line. The reference power offers the most power for the UMSI chip and sensor(s), however it can be shut off by the control electronics when the system is in a low power mode. The constant power may be used by circuitry which monitors passive sensors (e.g., switching

devices) and will remain powered even when the system is in a low power mode. Further power management of the UMSI chip is provided by having many of the circuit blocks on separate power lines which can be selectively connected (via wire bonds) to enable only the portions of the chip which are needed for a given application.

3.1.3 Application Adaptability

The UMSI chip supports a variety of transducer types via an analog interface which provides readout of capacitive, resistive, and voltage output devices. This allows the UMSI chip to be employed in a variety of applications where the transducers can be selected to meet the specific needs of each application. Furthermore, the UMSI chip support plug-n-play which allows sensor modules to be added, removed, or updated from a microsystem without affecting the integrity of the overall system. Primary features of the UMSI which support plug-n-play and application adaptability are the use of a standard bus for interface to system control electronics, support of expansion via the external SPI interface, and the support of several working modes such as IEEE P1451 standard interface mode vs. Mixed-signal interface mode and Single Module mode vs. Multi-Module mode. These modes provide not only adaptability to current applications, but also options for different usages in future systems.

3.1.4 Design for Future Expansion

The UMSI design contains a considerable amount, if not all, of the interface electronics needed for a microsystem application on a single chip. This chip can be marketed as a stand-alone microsensor interface solution or as a component in a complete sensor module. Furthermore, the chip provides the flexibility to expand into unforeseen applications by providing a programmable link to external devices such as high-accuracy

A/D and D/A components and/or large volume EEPROMS, all of which can be accessed through the UMSI chip via the sensor bus.

3.1.5 Interrupt Generation and Handling

A separate digital signal in the physical interface is provided to allow the UMSI to request service from the Controller. This interrupt signal is used in conjunction with the UMSI chip Status Register and Interrupt Mask Register to indicate exceptional conditions have occurred. When servicing an interrupt, the Controller should always read the status register which will specify the interrupt sources. The mask register is used to disable some of the interrupt sources. The interrupt can signal the Controller during normal operation to respond to the interrupt. During times when the Controller is operating in a low-power sleep mode, the UMSI interrupt will wake the system when sensor detected conditions warrant immediate attention. Examples of this include a physical shock or other sensor signal over a preset limit being monitored by the sensor module(s). This feature is especially important for very low power systems where it is expected that the system will spend most of its time in sleep mode to save battery energy.

3.1.6 Triggering

Signal triggering allows the Controller to send a timed gate signal to the sensor front end via the NTRIG signal lines on the physical bus. This allows actions within the UMSI chip to be timed and synchronized by the Controller. The trigger signal is applied to all nodes on the bus at the same time. A Trigger Enable control bit in UMSI memory determines whether the trigger signal will be passed through that specific chip. The trigger signal is utilized by the UMSI chip to activate and stop the temperature sensor counter thus providing an accurate gate timing to control readout of this sensor.

3.1.7 SPI port

The UMSI chip includes a SPI port to interface to external components to extend the functions of the UMSI. Serial EEPROM can be used to store the TEDS including the UMSI chip ID. An external serial A/D can be used to sample the analog sensor signals and provide sensor module output in a digital format. A serial D/A can be used to generate analog voltages used as set points for analog circuitry within the UMSI or as input to actuator devices. The SPI interface gives the UMSI the flexibility to be used in many applications which are not directly supported by hardware on the UMSI chip.

3.1.8 Retrieve Chip ID from EEPROM

Each time an UMSI chip is powered up it needs to load the chip ID from I/O port or EEPROM to identify itself among the different chips on the IM^2 Bus. To retrieve the chip ID from external memory, the Controller uses the "LDID_ROM" command (using the broadcast address: 0xFF) to load the ID from EEPROM. This operation, which is shown in Figure 3.2 is similar to the "TII_SPI" operation; however, no DOUT signal is enabled from the DOUT pin. The "LDID_ROM" instruction, unlike the "TII_SPI" operation, requires the UMSI to monitor the SPI SO signal (from EEPROM) in order to capture the ID data. This requires the UMSI chip to count cycles of SCLK in order to determine exactly when the ID information will be on the SO line. Finally the ID data is saved in an UMSI register and used as the preset chip ID for the UMSI chip. Notice that this command allows the chip ID of all chips on the bus to be retrieved at once, using only one instruction from the Controller.

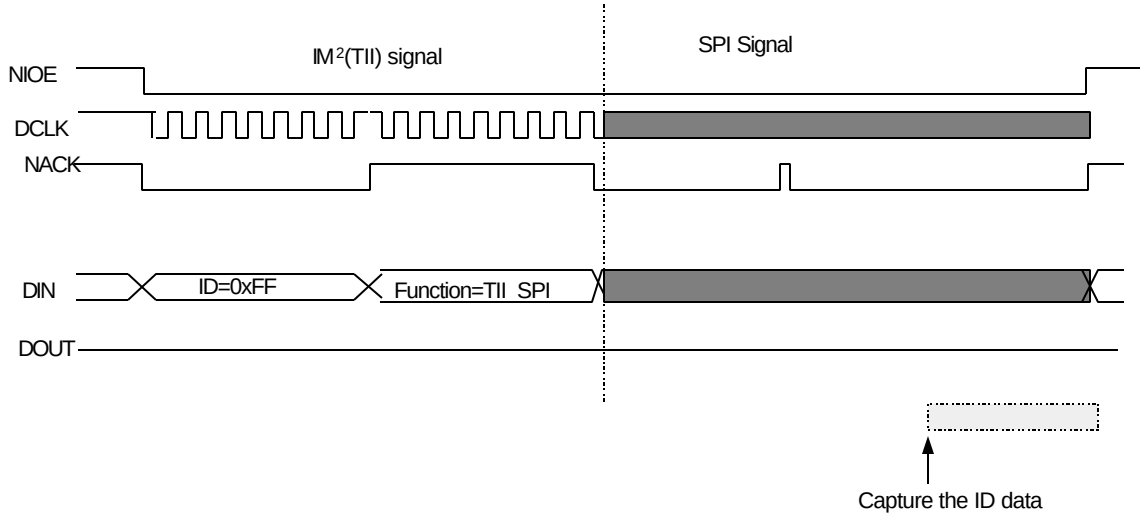


Figure 3.2: UMSI gets the ID from EEPROM

3.1.9 New Node Detection

NSDET is also driven by the STIMs as a way to signal STIMs' ID loaded and the new STIM presence. It might be pulled up on the host when no new node adds into the IM^2 Bus. NSDET will be pulled down when all or any node is powered up or re-powered up. Most time, the host may work in sleep mode and close the power of STIMs. When STIMs are powered up again, each STIM will pull down the NSDET until the host sends the command to ask them to load ID. Load ID has two cases: load ID form IO port and load ID from EEPROM (TEDS). At the case of load ID from IO, ID is wired bond to IO port by user. The host scans the whole possible Ids to find which is occupied. At the case of load ID from EEPROM, EEPROM of new STIM will be blank. When a new STIM is added to the bus, the NSDET will not be restored to pulling up situation until the host allocates a new ID. The new ID will be saved to EEPROM. NSDET is used as a plug-n-play indicator and help the host to manager and allocate the ID.

When any new node is added into the IM^2 Bus or system powers up, the NSDET will be pulled down to tell Controller that nodes need to load ID. Controller needs to broadcast LDID_IO or LDID_ROM command to load ID saved in IO port or EEPROM. After loading the ID data, most time the NSDET request will be cancelled by the UMSI chip. But, to loading ID from EEPROM, if the ID in EEPROM is blank (new node), NSDET will be cancelled until Controller allocates a new ID. ID in EEPROM is composed of 1 status bit (ID_status) plus 7 ID bits. If ID_status bit is "0", that means no valid ID allocated for this node. A valid ID should be from 0 to 127 and ID_status should be "1".

3.1.10 TEDS

The Transducer Electronic Data Sheet (TEDS) allows a system to automatically load the calibration constants for a particular transducer and sensor. As an electronics datasheet, TEDS permits the host to find out what is on the other end of the wire—a sensor version of plug and play.

The EEPROM will be used to save the TEDS. IEEE standard has detail definition of the data structure of TEDS. The Controller can reads/writes the TEDS (EEPROM) to perform the Plug and Play function.

3.2 IM^2 Bus Interface

The IM^2 bus interface is a critical part of UMSI chip. Microcontroller sends instructions and receives information all through bus interface circuit. The Sensor Bus Interface Circuit includes several circuit blocks:

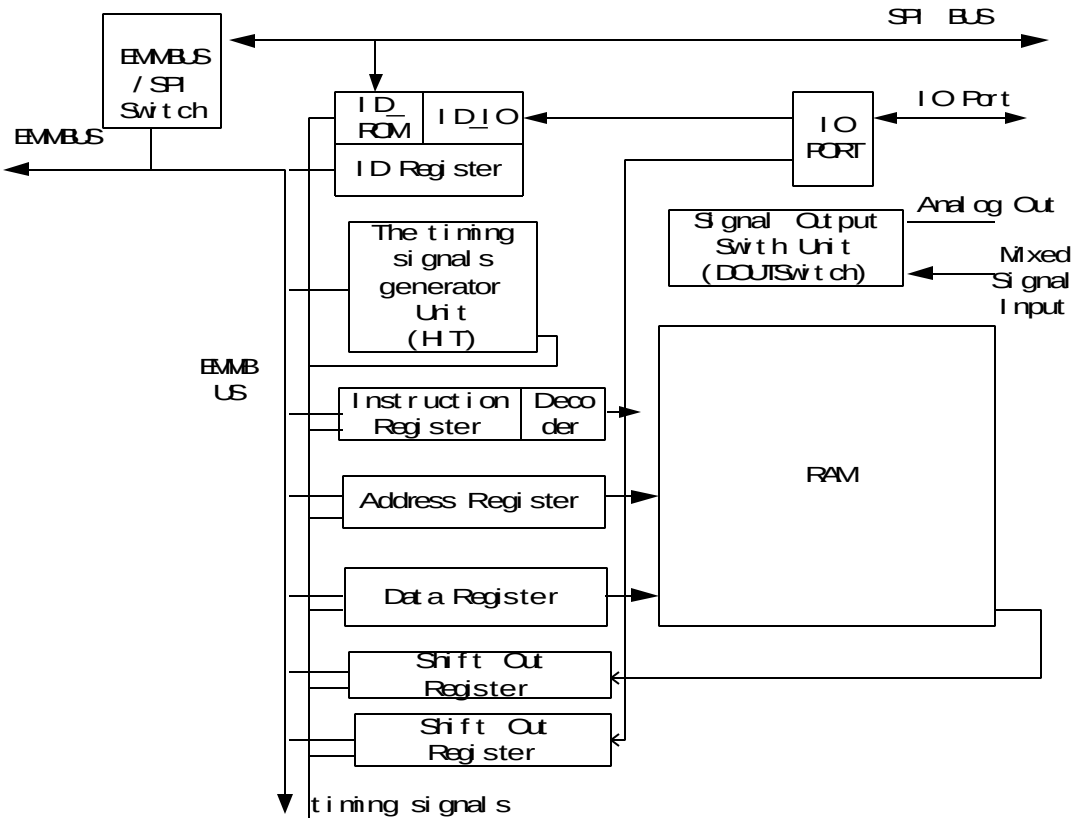


Figure 3.3: Overview of UMSI interface circuit

- ID register: loading ID from IO port or EEPROM and comparing it with the ID data from CEM
- TII/SPI block: switching the EMMSB signal to SPI signal when CEM need to communicate with SPI components
- Timing Signals Generator Unit: generating the time control signals to other blocks.
- Instruction register and decoder: receiving the Function data from CEM and decode it.
- Address register: receiving the Address from NACP
- Data register: receive the Data from CEM.

- RAM is two 8X8 bits latch units to offer programmable control signals to other circuits. Two Shift Out Registers are used to read out the data in RAM and IO port.
- IO port is a mutli-function port. It can be used as normal input & output ports or ID wired bond or SPI select.
- Signal Output Switch Unit is used to select one channel of analog signal to send out or sample it via external A/D.

3.2.1 Instruction register and decoder

The UMSI utilizes an 8-bit shift in instruction register. Only 4 bits is used for decoding. The maximal number is 16 functions. Now UMSI includes 9 functions totally.

Table 3.1: Instructions set for UMSI chip

Instruction Name	Function Byte Code	Operation
ReadMem addr,data	xxxx0000b	Read RAM data from UMSI
WriteMem addr,data	xxxx0001b	Write data to RAM
ReadIO addr	xxxx0010b	Read I/O
WriteIO addr, data	xxxx0011b	Write I/O
LdID_IO	xxxx0100b	load ID from I/O
LdID_ROM	xxxx0101b	load ID from EEPROM
RdSensor	xxxx0110b	Read Internal Sensor in the UMSI chip
OutMixed	xxxx0111b	Send the mixed signal to Dout
TII_SPI	xxxx1111b	Transfer data with SPI bus

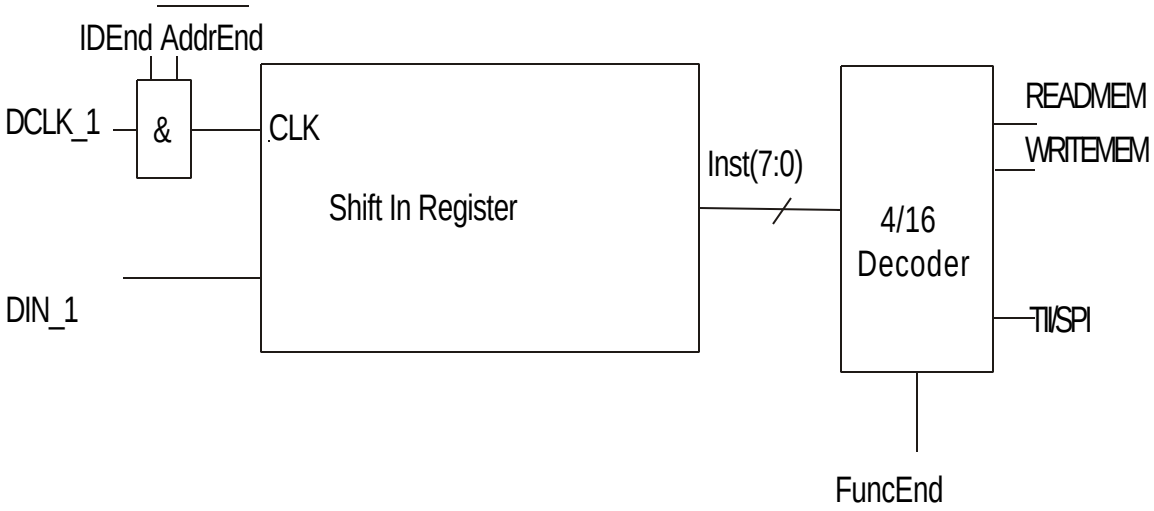


Fig 3.4: Instruction register and decoder

3.2.2 The Address Register and Write in Data Register

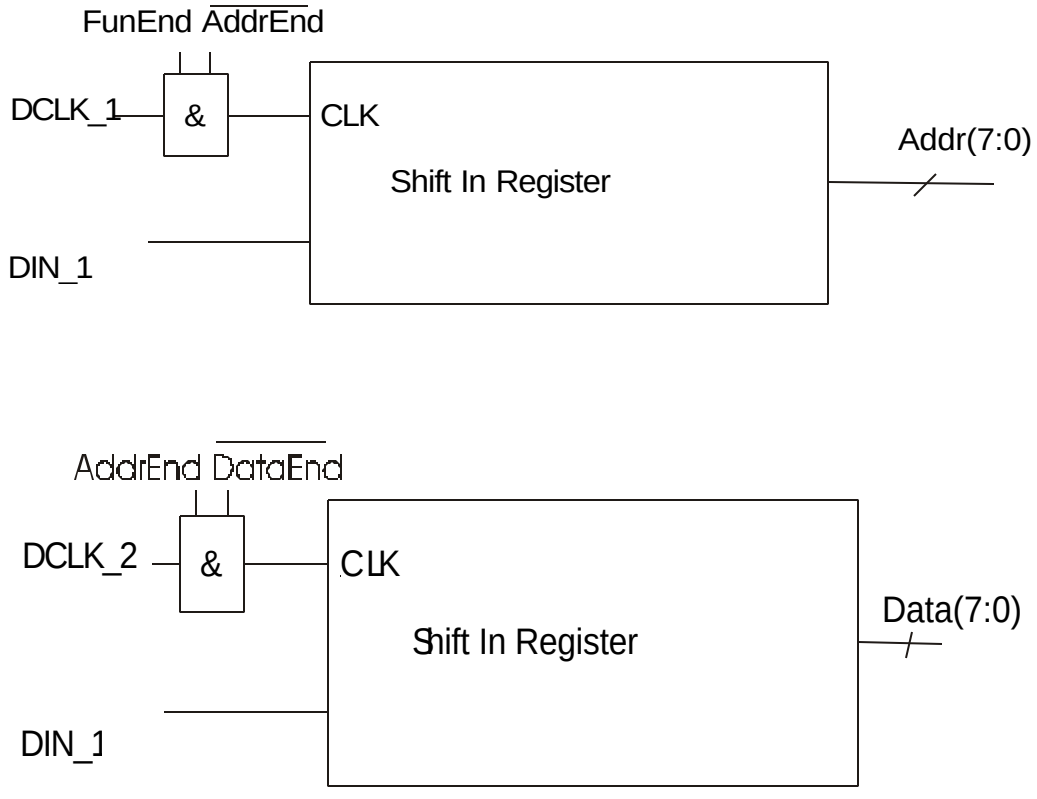


Figure 3.5: The address register and write in data register

3.2.3 The ID register and compare it with ID set in EEPROM and IO

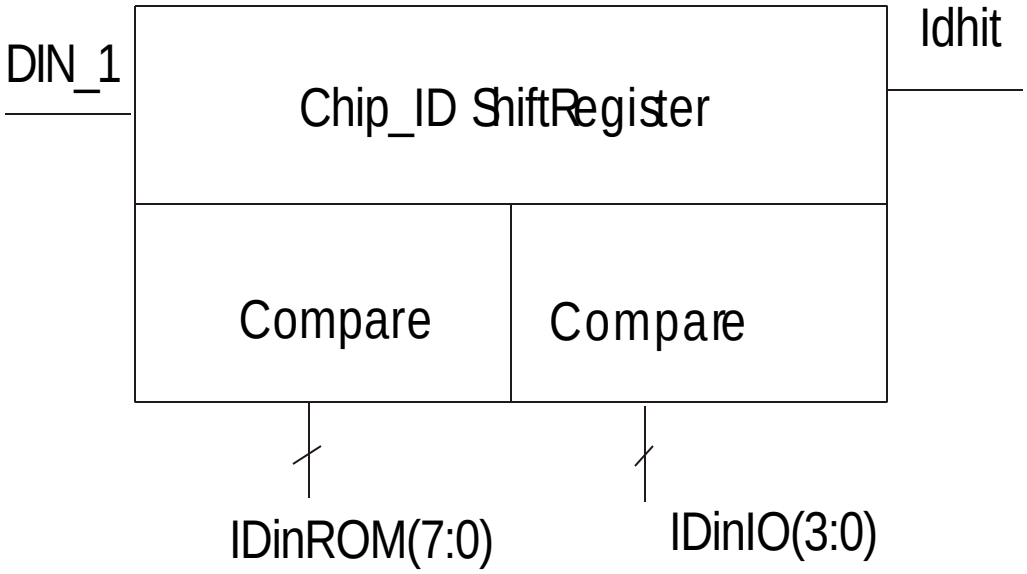


Figure 3.6: The ID register and compare it with ID set in EEPROM and IO

ID register save the ID called by the EMM and then match it with the preset ID from EEPROM or IO. If it is matched, then a matched signal Idhit will be generated.

The following circuits are used to load the preset ID saved in EEPROM and IO port.

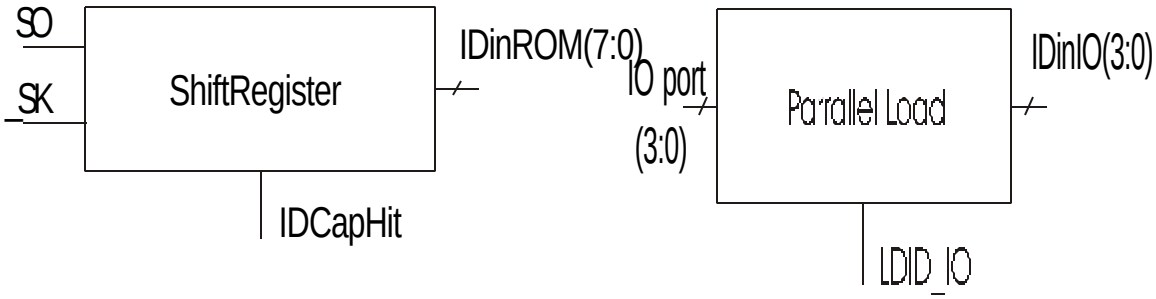


Figure 3.7: The ID register and compare it with ID set in EEPROM and IO

3.2.4 The signals switch between EMMSB and SPI bus

Switch the EMMSB signal to SPI signal when CEM need to communicate with SPI components.

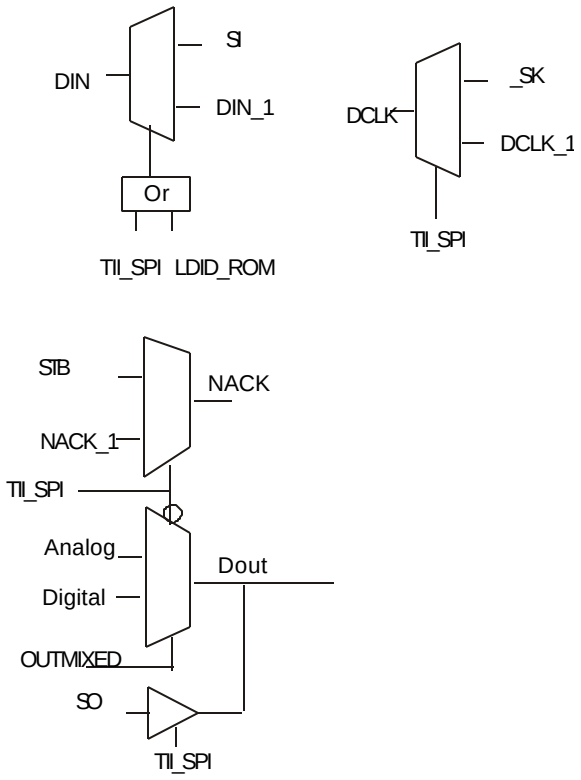


Figure 3.8: The signals switch between EMMSB and SPI bus

3.2.5 Data (Signal) Output Path

DOUT is the final pin for UMSI chip to send out the Data (Signal). There are three types of signal: Digital Signal, Analog Voltage Signal and Frequency Signal. The Digital Signal is shifted from the RAM, Sensor Counter Register (such as Temperature Sensor) and SPI bus. The Analog Signal can be directly from sensor, amplifier. Some sensors sent out frequency pulse signal. The frequency is relation to the physical parameter.

The Analog/Frequency Signal Path

The analog signal from sensor or frequency signal can be linked to Analog0~7. One channel of them will be selected to Analog output pin and at the same time it will be routed to Dout when CEM send "OutMixed" function. Analog output pin links to the SPI A/D chip, so the signal will be sampled and data will be send back via a SPI read protocol. The channel select switch is controlled by Analog Select and SPI Chip Select Control Register.

The Digital Data Path

The Digital Data sources include the memory read out register, IO read out register and sensor read out register (such as temperature sensor). Which channel will be shift out is decided by the function code.

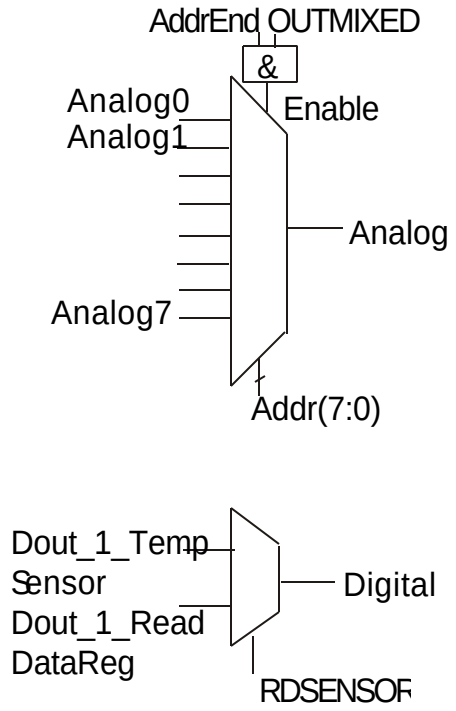


Figure 3.9: Data (Signal) output path

3.2.6 The memory and IO read out register

Two Shift Out Registers are used to read out the data in RAM and IO port.

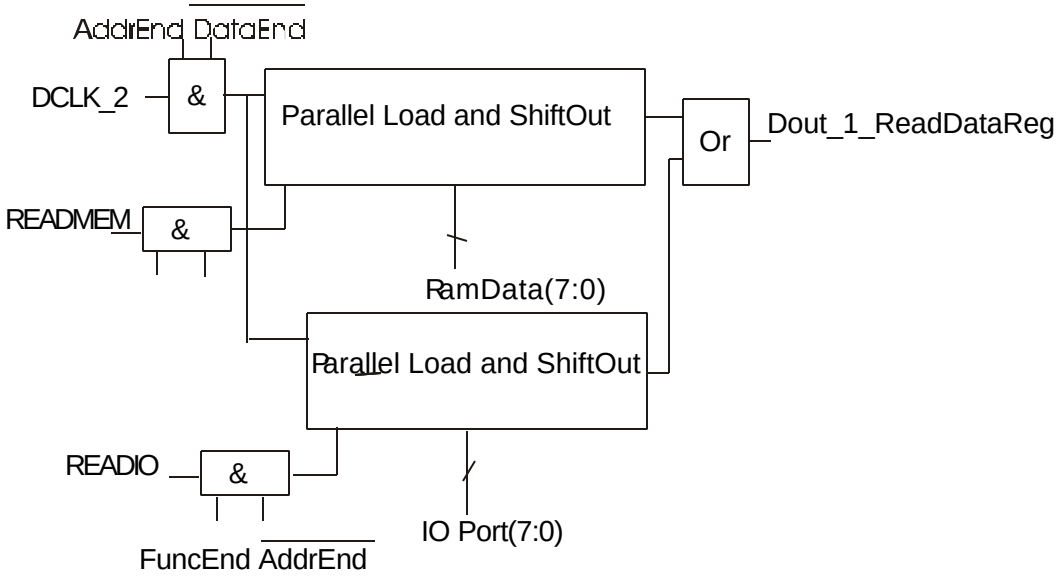


Figure 3.10: The memory and IO read out register

3.2.7 IO port

IO port is a bi-direction port. It is used as a read port when UMSI need to read a ID from IO port or UMSI performs the function of "ReadIO".

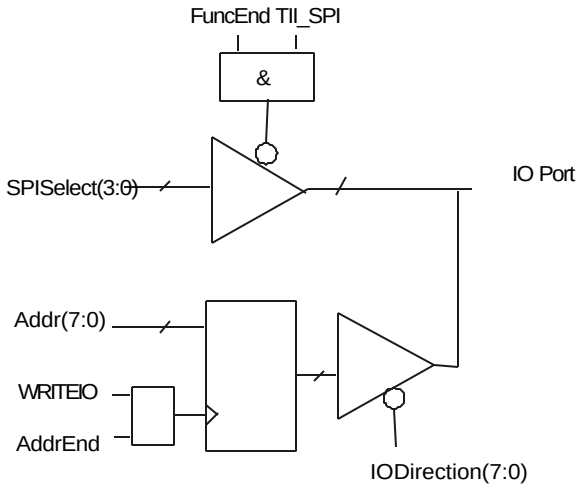


Figure 3.11: IO port

As a write port, firstly, it is used as the SPI chip select. When the CEM sends the "SPI_TII" function, the TII bus will run the SPI signals. At the same time, UMSI need to offer a SPI chip select signal to choose the target SPI chip. Secondly, IO port can be used as normal IO when CEM sends the function of "WriteIO". IO Direction Control Register is used to enable the output.

3.2.8 Timing Signal Generation

The timing signals are very important inner signals just like the engine to the car. The timing signals make other circuit to do some tasks in a designed sequence. At here, IDCAPHIT is a gate siginal for capturing the ID saved in EEPROM; IDEnd is a rasing edge signal which is activated when the first falling edge clock signal comes after the ID. IDEnd signal will be return to 0 after the NIOE=1. It is similar to FuncEnd, AddrEnd, DataEnd signals. WriteMemTrig is used to write data to the memory. It is a pulse signal which width is same as the DCLK.

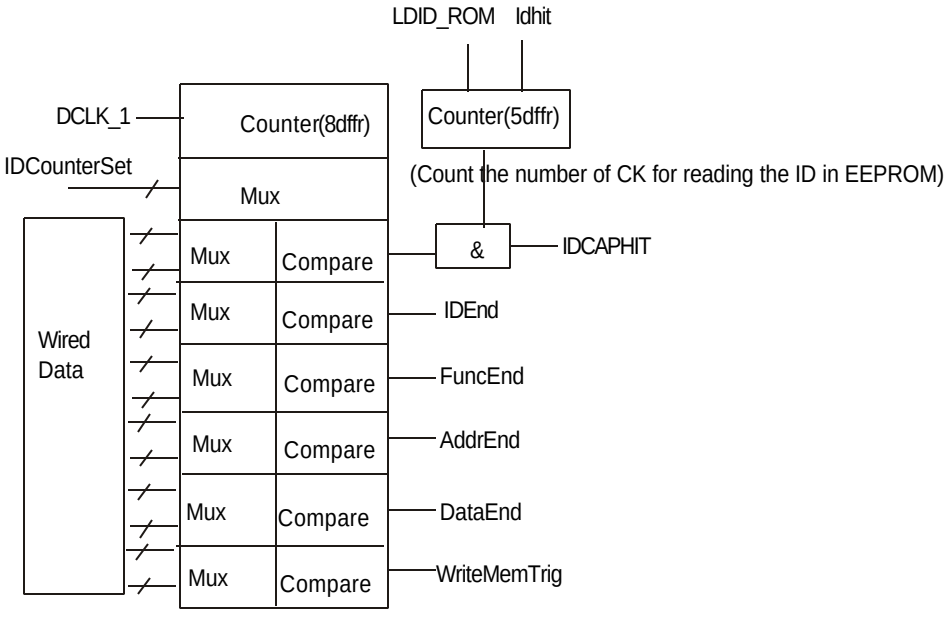


Figure 3.12: Timing signal generation circuit

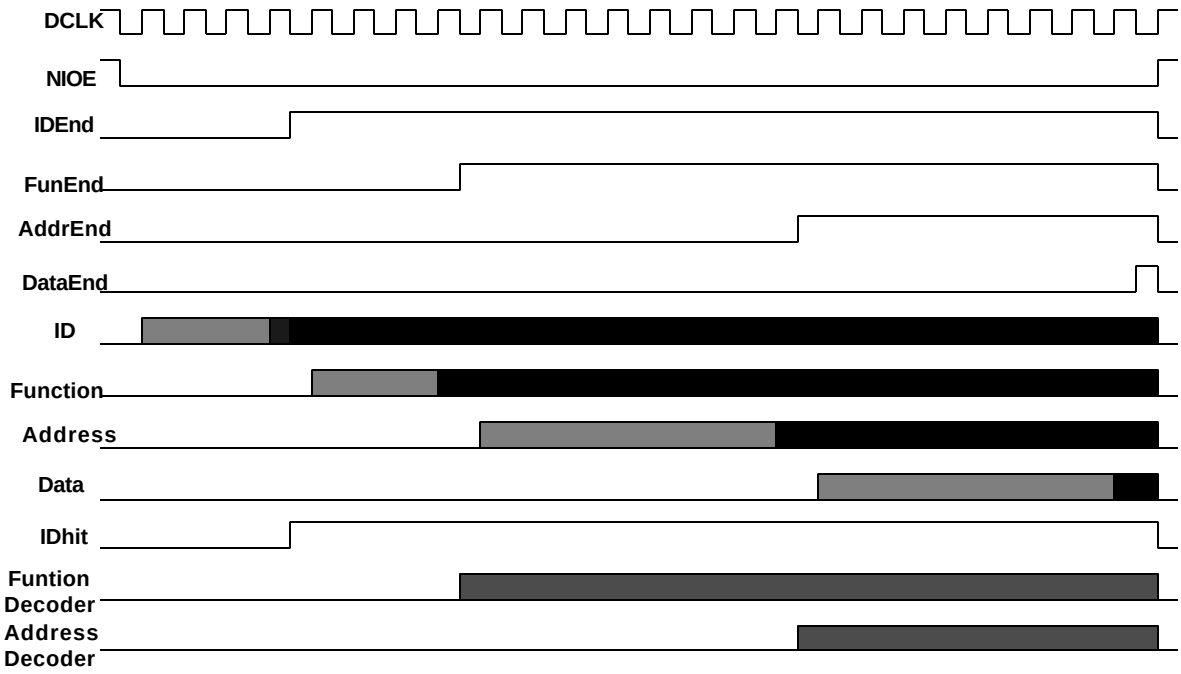


Figure 3.13: Internal signals timing

3.2.9 Power Supplies

According to the demand of EMMSB, two powers are need to UMSI at least. One is from reference voltage that can be closed by the CEM, so called it Controllable Power. Most part of UMSI and sensors will use this power. When system works in sleep mode, it will be closed to minimize the power consumption. Another is Constant Power. It is used for the part that need continual power supply. Shock sensors are used to detect the emergency event and wake up the whole system from sleep mode. So shock sensors and the wake-up logic should be link to constant power.

We also consider separating the power supplies of each import part, such as Temperature sensor, Analog amplifier. We offer the independent power to those parts to make easy to test which part is working, which part is wrong. Finally, we will link them together to the Controllable Power after the test and no problem exist in the chip.

3.2.10 The Final Schematic

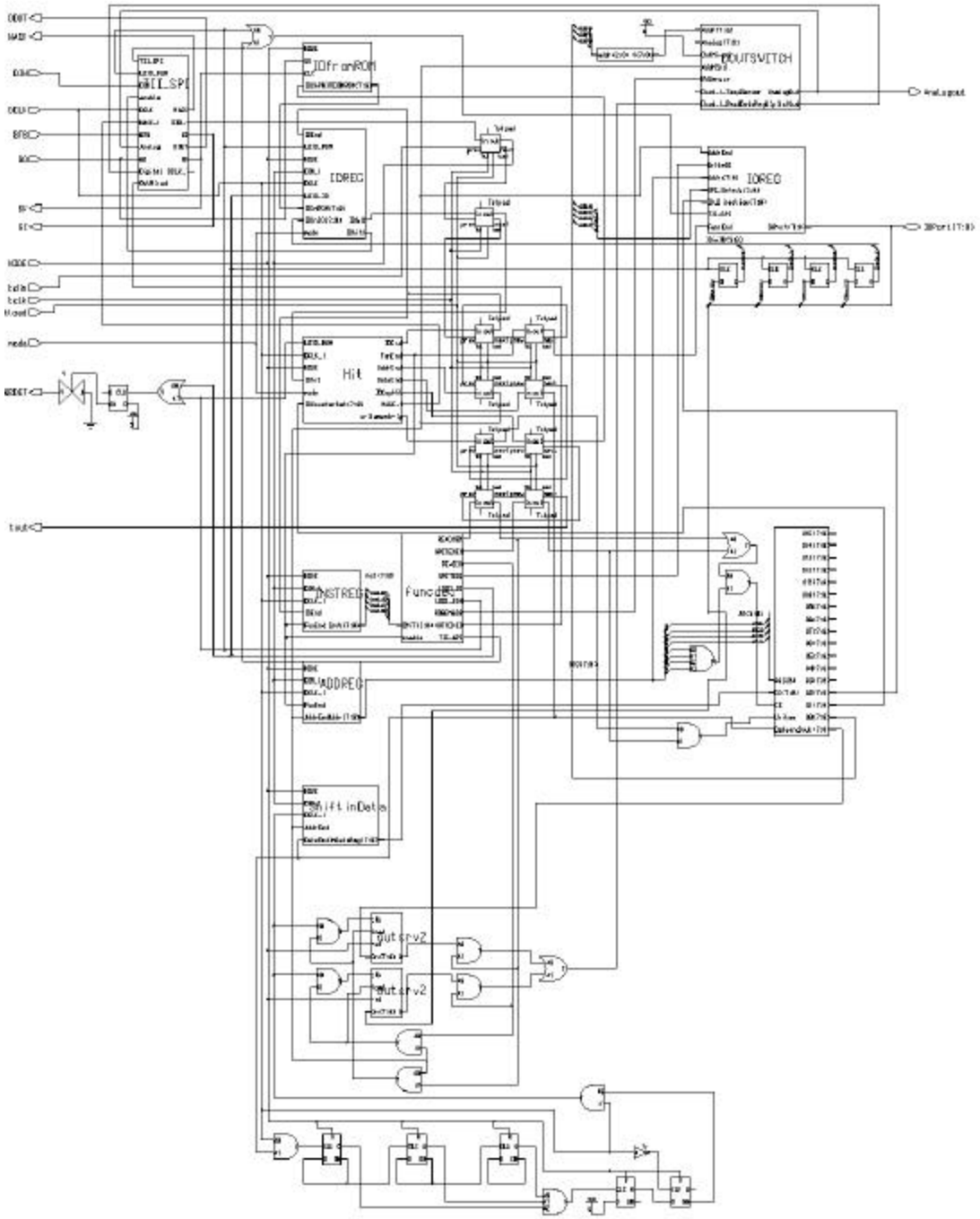


Figure 3.14: Schematic of UMSI bus interface (Top Level)

3.3 On-chip SRAM and Status-readout circuit

SRAM comes in many different formats and styles. The type of SRAM that is preferable for a given application is a function of the required memory size, the time it takes to access the stored data, the access patterns and the system requirements. In general, the circuit of SRAM includes memory array, decoder, control circuit, sense amplifier, read/write circuit and is controlled by the clock signal. But the SRAM used in UMSI is relatively small and special, so we can design the SRAM in distinctive way.

Requirements:

1. The SRAM stores the commands that control other parts of the circuit, so each memory unit should has its own output ports to realize controlling when system requires.
2. Sometimes status of some parts of circuit needs to be readout. This function is also included in SRAM to share addresses and share readout circuit.
3. In some case, system would test the data stored in SRAM, so a general data output bus is required to output data from any memory unit of SRAM.
4. Because it is built-in memory, all control circuit is from other part of circuit; control circuit is not employed. Because of its small size, it is unnecessary to use sense amplifier and column decoder.
5. The SRAM can be extended.

To implement the function of this kind of circuit, two memory blocks are introduced. One is a 16-byte memory where each byte is 8 bits wide, another is a combination of 8-byte memory and 4-byte status-readout circuit, as shown in Figure 3.15.

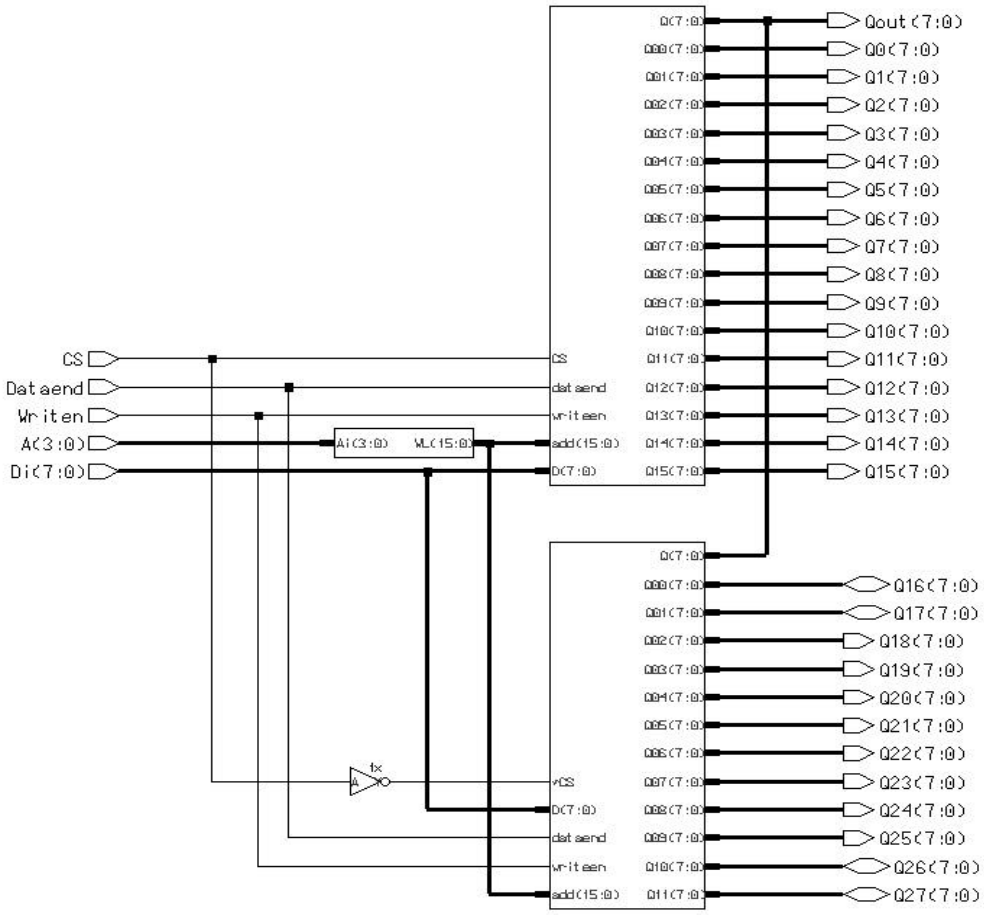


Figure 3.15 On-chip SRAM schematic

The following is list of inputs and outputs,

[Ain (3:0)]: 4 bits address signals to select a memory unit from one SRAM block.

[dataend]:a control signal to notify SRAM that data has already been prepared on data bus.

[writen]: a control signal to make SRAM step into writing operation when data is ready.

[CS]: block-select signal to make SRAM block active or inactive.

[Din (7:0), Qout (7:0)]: separated input/ouput data ports.

[Qn(7:0), n=0, 23]:data output from each memory unit of SRAM to control correspondent circuit of the UMSI system(This is special requirement in UMSI system).

[Qn(7:0), n=24, 27]: data output form each status-readout circuit.

In this circuit, a block selecting signal is employed named CS. When it is effective, the block of SRAM can be read or written. Otherwise, the circuit is 'inactive'. When several blocks of this kind of SRAM are used in system, we can set one active and others inactive only by making CS available or unavailable. So we can extend the size of SRAM in UMSI system if necessary. This approach has a dual advantage.

1. Even though the size of SRAM is relative large, the access time is relative small.
2. The block address can be used to activate only the addressed block. Inactive blocks are put in power-saving mode with decoders disabled.

Table 3.2: The truth table of the SRAM block

CS	Written	Dataend	Ain(3:0)	Din(7:0)	Qout(7:0)	Qn(7:0)	Operation
0	1	1	m=0,15	Y	Y	Qm(7:0) =Y	Write
0	0	1	m=0,15	Y	Y	Qm(7:0) =Y	Write
1	1	1	m=0, 7	Y	Y	Qm_{+16}(7:0)=Y	Write
1	0	1	m=0, 7	Y	Y	Qm_{+16}(7:0)=Y	Read
1	1	1	m=8, 11	Y	Y	Qm_{+16}(7:0)=status	Status readout

3.4 Analog Interface Circuit

The Figure 3.16 shows the architecture of the UMSI chip analog interface. This circuit can read out capacitive sensors, resistive sensors, and sensors with direct voltage

output. It is highly programmable and provides offset and gain adjustments. Also, it supports self-test for physical capacitive sensors.

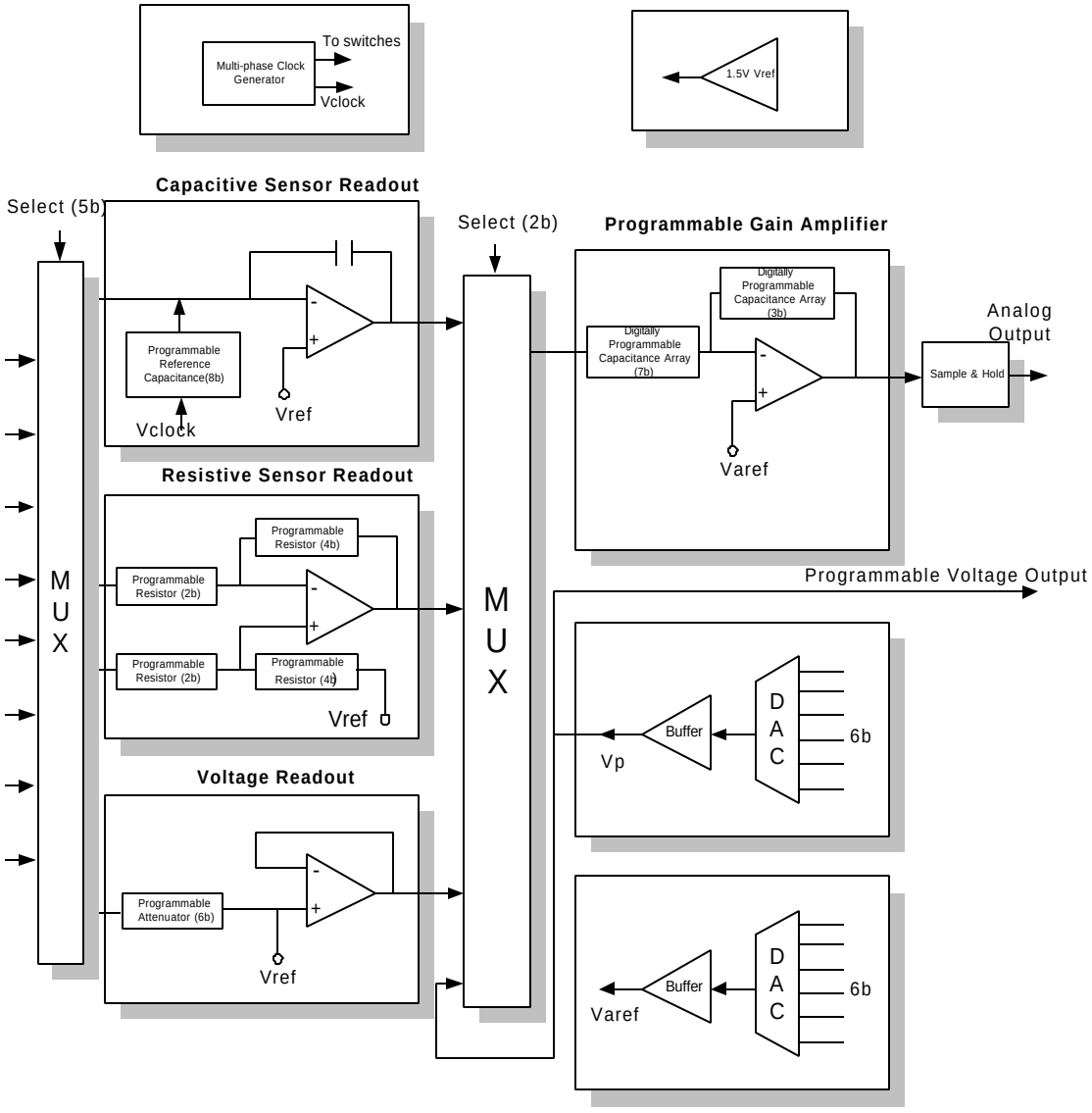


Figure 3.16: Architecture of the UMSI chip analog interface

Through 8-to-1 input multiplexer, the circuit can interface with up to 8 sensors that are capacitive or/and resistive or/and voltage-output. Output voltage of capacitive readout is proportional to the difference of capacitance between off-chip capacitive sensor and an on-chip programmable reference capacitor. The resistive sensor readout

interfaces with a resistive sensor half or full bridge, and provides an output corresponding to the bridge resistor change. The voltage readout forces the input voltage to a desired range. Three outputs of these readout circuits go into a 4-to-1 multiplexer with self-test signal. One of them is selected and amplified by gain stage, whose sensitivity can be programmed variously. And finally, the output from gain stage is stored on capacitor in the sampled and hold stage.

Additionally, there are two 6-bit digital to analog converters (DAC). One is used to generate voltage Varef applied to programmable gain amplifier to offset output signal present in the absence of any input signal. Another is to generate voltage Vp for self-test of the physical capacitive sensor. The Vp drives the sense and reference capacitors in each input charge integration cycle. It can be used to apply a programmable DC voltage and electrostatic force to the sensor for self-test. In self-test mode, the programmable electrostatic force is used to move a sensor structure and generate a change in the sensor's output. An analog ground generator is used to generate 1.5V analog ground applied to capacitive, resistive, and voltage readout circuits, and sampled and hold stage to cancel any offset. Multi-phase clock generator is to get non-overlapped clock phases to control switches in circuit.

3.4.1 Capacitive readout, programmable gain stage and sample & hold stage

Figure 3.17 shows how to interface capacitive readout, programmable gain stage and sample & hold stage. Cs is the sense capacitor, and Cref is the reference capacitor that is controlled by 8-bit of SRAM. During phi1, the reset switch of the charge integrator is closed and Cs is charged through the charge integrator output to Cs(Vref-Vdd), and Cref is charged to Cref(Vref-Vss). Once phi1 goes low, the charge stored in Cs changes

to Cs(Vref-Vss) and that in Cref changes to Cref(Vref-Vdd). The net change in charge is (Cs-Cref)(Vdd-Vss). This change of charge is transferred to feedback capacitor (Cf1). The magnitude of the output voltage will be equal to (Vdd-Vss)(Cs-Cref)/Cf1. So the output voltage is proportional to the difference between the capacitance of the input sense capacitor and that of the reference capacitor.

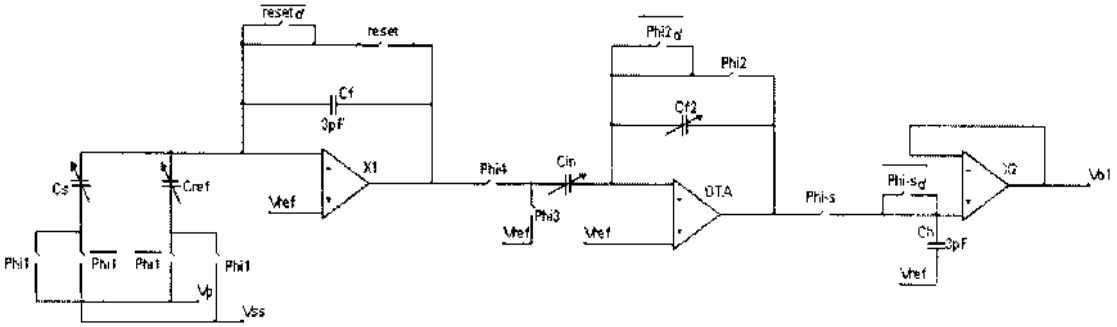


Figure 3.17: Capacitive readout, programmable gain and sample &hold stage

Subsequently, phi2 goes high, the second stage integrator is reset and Cin that is controlled by 7-bit clock phases from clock generator is charged through the charge integrator output. Cin is charged to Cin(Vout1-Varef). Clock phases phi3 and phi4 are slightly delayed phi1 and phi2 clocks, respectively. After phi3 goes high, the voltage across Cin changes to zero, and hence the change in charge stored in Cin is equal to Cin(Vout1-Varef). The gain of the second stage is determined by the ratio of the total capacitance switched into its input to the feedback capacitance (Cf2), that is, (Vout1-varef)Cin/Cf2.

Finally, during phis (same with phi3), the output voltage of second stage is sampled and held onto capacitor Ch until next phase of phis. During this time the output of the sample and hold equals the last output of the programmable gain stage. The

dummy switches controlled by reset_d_bar, phi2_bar, and phis_bar are used to reduce clock-switching noise at the high impedance nodes. The circuit is designated to operate with a 50-kHz clock. The non-overlap time of the clock phases and the clock delays are both 250ns.

3.4.2 Resistive readout and voltage-output readout

The resistive sensor develops a voltage that varies with resistance by using a resistive full bridge, which converts an imbalance in resistor values to a voltage. The schematic of resistive readout is shown in Figure 3.18. The bridge output voltage is applied to the input of a closed loop differential amplifier. The gain of this amplifier is given by the ratio of R2 to R1. R2 and R1 are two programmable resistor controlled by 2-bit and 4-bit memory. A high gain of Opamp of the resistive readout is required to ensure precision operation; to accommodate a large output signal, large output voltage swing is required; and the output needs to be buffered to drive current into a resistive load.

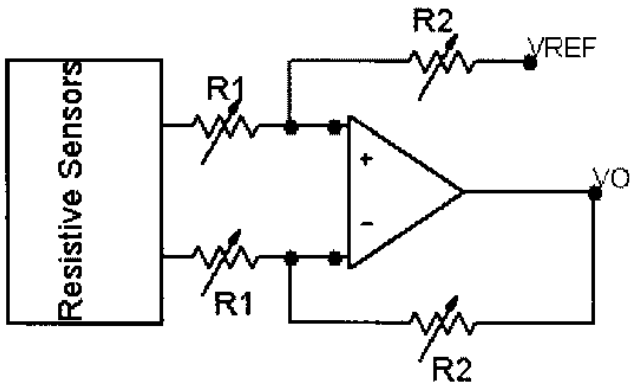


Figure 3.18: Resistive sensor readout

By attenuating the input voltage range and feeding it to the programmable gain amplifier stage, the voltage-output readout provides control over the voltage range. The

schematic of the voltage readout is shown in Figure 3.19. The input voltage is attenuated by a programmable attenuator that is controlled by 6-bit memory and is fed into the non-inverting input of the Opamp. The very high gain of the Opamp forces the voltage at the inverting node to be equal to the voltage at the non-inverting node. The function of the Opamp is to provide buffering.

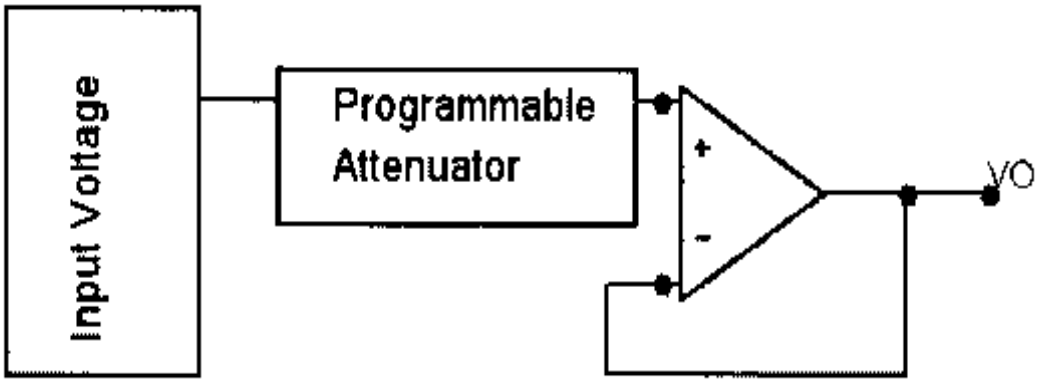


Figure 3.19: Voltage-output readout

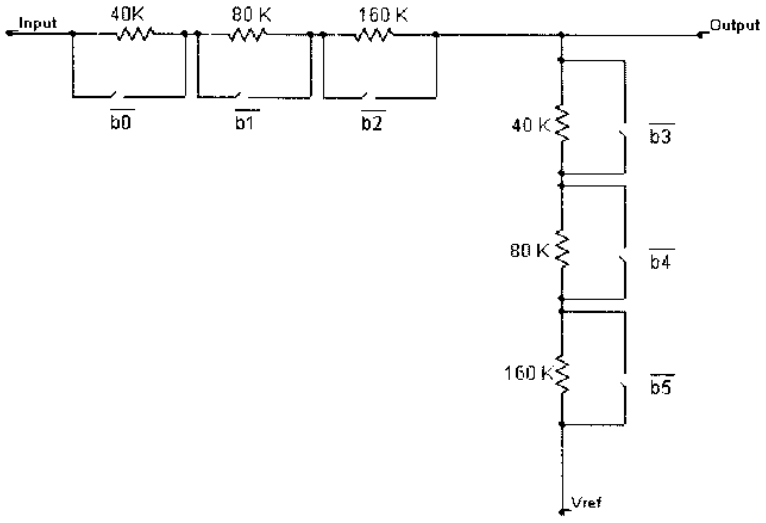


Figure 3.20: Programmable attenuator

3.4.3 Digital-to-Analog Converter (DAC) and Analog Ground generator

Figure 3.21 shows the schematic of the 6-bit DAC. This circuit converts an input digital signal represented by 6-bit memory to an analog output. 6 bits set the voltage at the non-inverting node of the Opamp by directing the current (*i*) from the respective current source to flow through the R1. The voltage at the non-inverting node of the Opamp is an *i*R1 drop below the positive power where *i* is the total current flowing through the resistor R1. That is this voltage is proportional to current *i.* The relation between the input digital codes to the output analog voltage is inverting, which corresponds to a linearly decreasing output voltage as the input code is incremented. The very high gain of the amplifier forces the voltage at the inverting node and hence the output voltage to be equal to voltage of the non-inverting node. As a result the output voltage is proportional to current *i.* The function of the Opamp is to provide buffering. The current source Ibias is used to set the output voltage that results when all switches to current sources. This voltage is set so that the voltage range for the DAC is symmetric around analog ground.

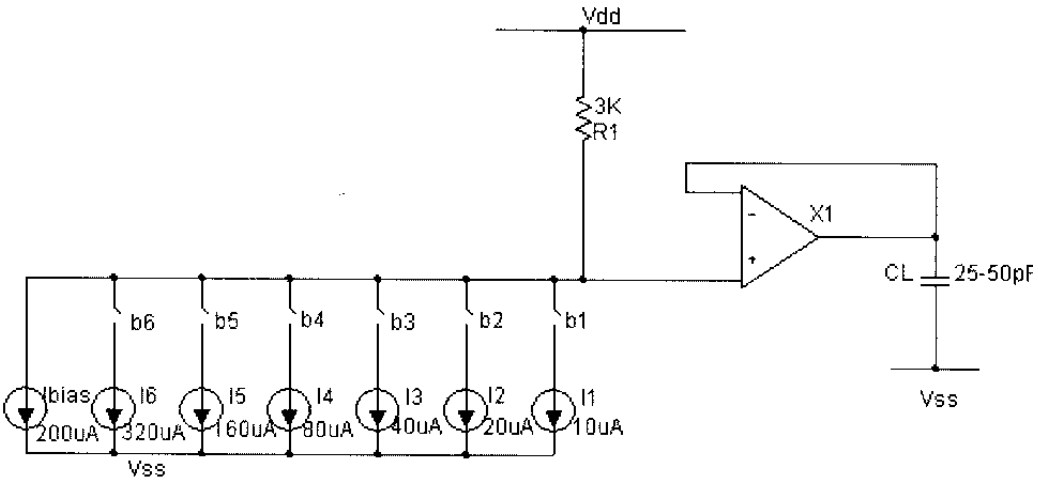


Figure 3.21: 6-bit digital-to-analog converter

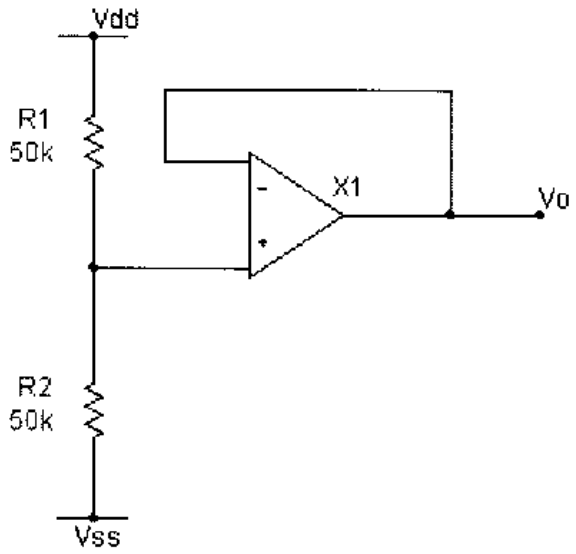


Figure 3.22: Analog ground generator

The buffer used for the DAC can also be used to generate the analog ground signal. The voltage between the positive power supply and ground is divided by two. The divider is formed by two resistors R1 and R2 as shown in Figure 3.22. This voltage is applied to the non-inverting terminal of the Opamp. The very high gain of the Opamp forces the voltage at the inverting node to be equal to the voltage at the non-inverting node. The function of the Opamp is also to buffer the voltage.

3.5 SRAM Assignment

Microcontroller control the UMSI chip via sensor bus which stores programming codes in on-chip SRAM. Two SRAM blocks whose sizes are 16x8 and 12x8 are used in the final UMSI chip. Mem[16], Mem[17], Mem[26], and Mem[27] are status-readout bytes. Totally 22 bytes in memory block are used by sensor bus interface, interrupt circuit (shock sensor), temperature sensor, and mainly analog sensor readout interface.

Sensor Bus Interface:

Mem[0]: Analog Select and SPI Chip Select Control Register

Mem[1]: Preset of ID Capture Counter Register

Mem[2]: I/O Direction Control Register

Interrupt Circuit:

Mem[16] & Mem[17]: Interrupt Status (Read Only)

Mem[3]: Interrupt Masks

Temperature Sensor:

Mem[4] & Mem[19]: Temperature Sensor Control Register

Mem[26] & Mem[27]: Temperature Sensor Status

Analog Sensor Readout Interface:

Mem[7] Digital-to-Analog Register 1

Bit [6: 0], DAC digital input for reference voltage Varef

Mem[23] Digital-to-Analog Register 2

Bit [6: 0], input signal of DAC for self-test to generate Vp

Mem[8] Amplifier Channel Select Register

Bit [4: 0], 8-to-1 Multiplexer Select. It decides which sensor output will be read out.

Bit [6: 5], 4-to-1 Multiplexer Select. It decides which readout channel will be amplified by second gain stage.

Mem[12]: Reference Capacitor Array Switch Register

This register is used to control reference capacitor array Cref in capacitive readout circuit.

Mem[13]: Attenuator Array Switch Register

Bit [5: 0], It is used to control programmable attenuator b0~b5 in voltage readout circuit.

Mem[14]: Feedback Capacitor Array Switch Register & Resistor R1 Control Register

Bit [6: 4], It is used to control programmable feedback capacitor array in second gain stage.

Bit [3: 0], It is used to control resistor R1 in resistive readout circuit.

Mem[15]: Resistor R2 Control Register

This register is used to control resistor R2 in resistive readout circuit.

Mem[25]: Clock Phase Control Register

Bit [6: 0], It is used to generate 7-bit clock phase with clock generator to control input capacitor array Cin in second gain stage.

Chapter Four

Chip Physical Layout Design

Poor layout affects the area of the chip and its performance (speed, power dissipation, etc.), and ultimately it can affect the correct behavior of the chip. Although layout is mainly driven by two-dimensional, geometrical constraints, such as aspect ratio of the circuitry and physical location of the terminals, performance issues and considerations on design reliability influence layout, as well. One example is that poor layout can lead to very long interconnections which slow down the chip, and in extreme cases, laying out power lines on diffused resistors may generate arcs which permanently short the two layers. Given the large number of parameters involved, it is difficult to present layout techniques in a well-structured, "scientific" way. Usually, layout techniques are presented as a list of tips that, in most, but not all situations are indeed effective.

4.1 Layout Design Overview

For designing the Universal Micro-Sensor Interface (UMSI) chip, the AMI05 C5N 0.50 micron, Mixed-Mode, 3 Metal, 2 Poly, +3V process was selected. The design rules for this process, which are obtained from the MOSIS founding service, are SCMOS_SUBM. ?=0.30μm in AMI05 C5N and feature size is 0.60μm. The process utilizes 14 design layers, including N-well, Active, N-select, P-select, Poly, Poly2, Hi-Res-Implant, Contact, Metal1, Via, Metal2, Via2, Metal3, and Glass.

A full-custom design methodology was used for UMSI chip. Initially, a standard cell library was established, based on the considerations of area, delay, driving ability, etc., in order to place and route all basic blocks of the digital circuit properly. During this

part of the project, different standard cells with different layout profiles but the same digital function circuit were designed to provide cells that would fit into diverse circuit blocks.

Structured layout methodologies were employed in this circuit. These methodologies aim to increase productivity and decrease the chance of mistakes. Every block in the circuit was placed and routed based on its function as well. For example, a basic block represents a basic function, several such blocks were placed together to make up a higher level block which can realize a more complicated function. The structure of the chip layout directly matched the schematic cells.

To ensure the whole layout is as compact as possible, the shape of each functional cell was considered during physical design. At the same time, for the reason that the property of the circuit should be maintained well, positions of these blocks must be counted in advance to make some key wires wider and shorter.

Finally, the low power dissipation requirements, of the UMSI chip requires dedicated treatment in the layout of power and ground lines. UMSI is designed to work under two modes: a low-power sleep mode and a wake up operational mode. Thus, two power lines, “always on” and “wake up”, must be separated and which circuits should connect to which power line must be considered during floorplanning and routing. Furthermore, to maintain the noise margin and chip reliability at acceptable levels, the chip also implements separate analog and digital powers lines. There are two power lines used in analog circuit, one supplies power for capacitive readout, gain amplifier, output stage, DAC1 and clock generator, another supplies for resistive, voltage readout and DAC2.

Figure 4.1 shows the floor-plan of the UMSI chip before routing.

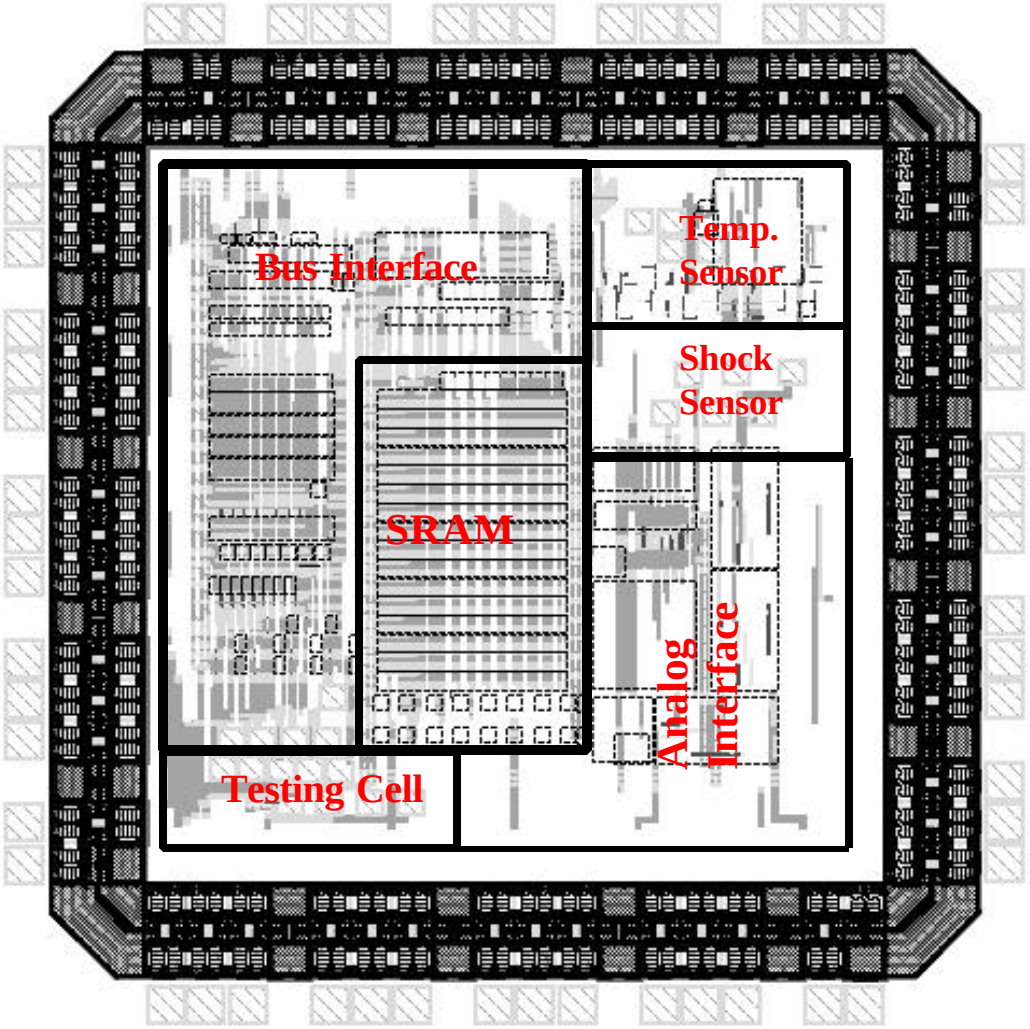


Figure 4.1 Chip floor-plan before routing

4.2 Standard Cell Library

Standard Cells are fundamental components in UMSI chip. It is very indispensable to optimize the cell library for circuit area, simulation accuracy, library simplicity and circuit speed. Requirements for accuracy and simplicity are usually conflict with those for functional density and circuit speed.

Among all these factors, chip design being correct should be emphasized on first time. Thus, simulations must be accurate. Full-custom chip designs must be simulated accurately to achieve higher performance. Every cell should pass DRC and LVS. And parameter extraction was performed on it. Finally, back-annotated schematic is used to do analog functional and timing simulation. Figure 4.2 shows the simulation result of D flip-flop with reset signal.

Simplicity, from layout designer's point of view, means he can easily find just the function required in the library, and that a dense layout is achieved automatically and easily, and is also important. Although, the requirement of speed for UMSI chip is not very harsh as described early, this standard cell library is not just for UMSI chip, it should also be worthy of further extension and use. Thus, speed factor is also considered during design. We do many simulations on every standard cell which drives 1 inverter, 4 inverters and 8 inverters. For a 2-input NAND gate, delay time are 2.1ns (rise)/2.4ns (fall), 3.4ns (rise)/3.5ns (fall), and 5.1ns (rise)/5.7ns (fall) respectively.

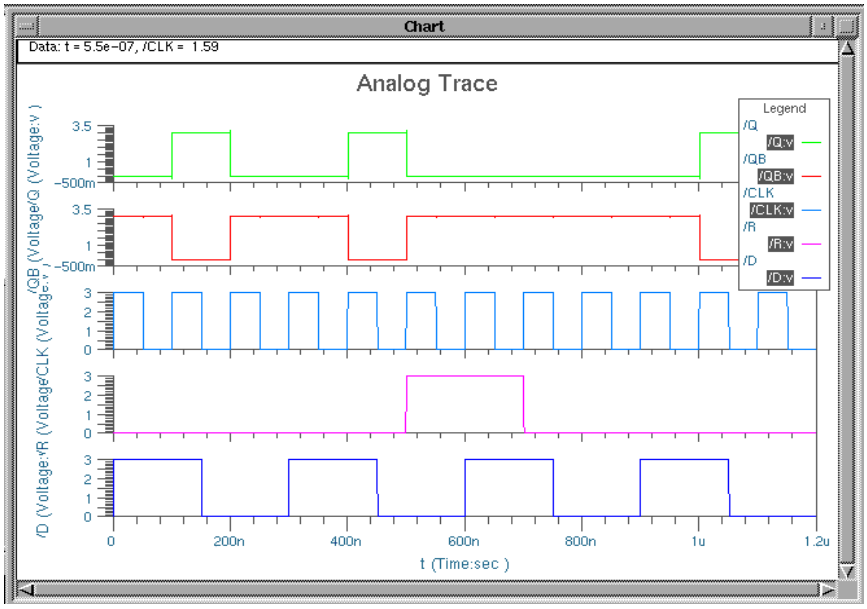


Figure 4.2 Functional and time simulation on D flip-flop with reset

Basic cell design parameters such as standard transistor sizes, power bus sizes, and the cell “foot print” were selected after a careful study of present and future chip requirements such as chip size, statistical routing lengths, long term reliability and performance. The standard transistor widths were chosen by finding the region of minimum delay and area for the most often used gates with a range of most likely loading. These fundamental parameters were maintained consistently throughout the cell library.

The standard ratio of Wp/Wn was selected to provide the minimum propagation delay in a range of statistical loads and was based on the effective transistor channel lengths and carrier mobility. A ratio of 1.8(9 lambda/5 lambda) was selected to produce near-optimum delays for both lightly loaded and heavily loaded outputs.

The gate capacitance of the standard transistor pair along with expected parasitic capacitance was used to determine the capacitance value of a unit load. The drive characteristics of a standard drivers (two input logic gates) driving a statistical load, provided standard transition times for excitation waveforms used to develop simulation models. The affects of both input loading and output loading are taken into account by the simulation model because long transition times on inputs of the cell affect the delay at the output.

Throughout the primitive cell library a major concern was minimizing the overall propagation delay while keeping input capacitance low and output drive high. All multistage primitives such as flip-flops, multiplexes and latches have buffered inputs and outputs. As seen in Figure 4.3 the D flip flop’s inputs and outputs are fully buffered which prevents output loading from changing internal set up and hold requirements.

Keeping the clock input capacitance low allows several flip flops to be clocked in parallel without on chip buffering which can produce hard to control timing problems such as clock skew and long transition times.

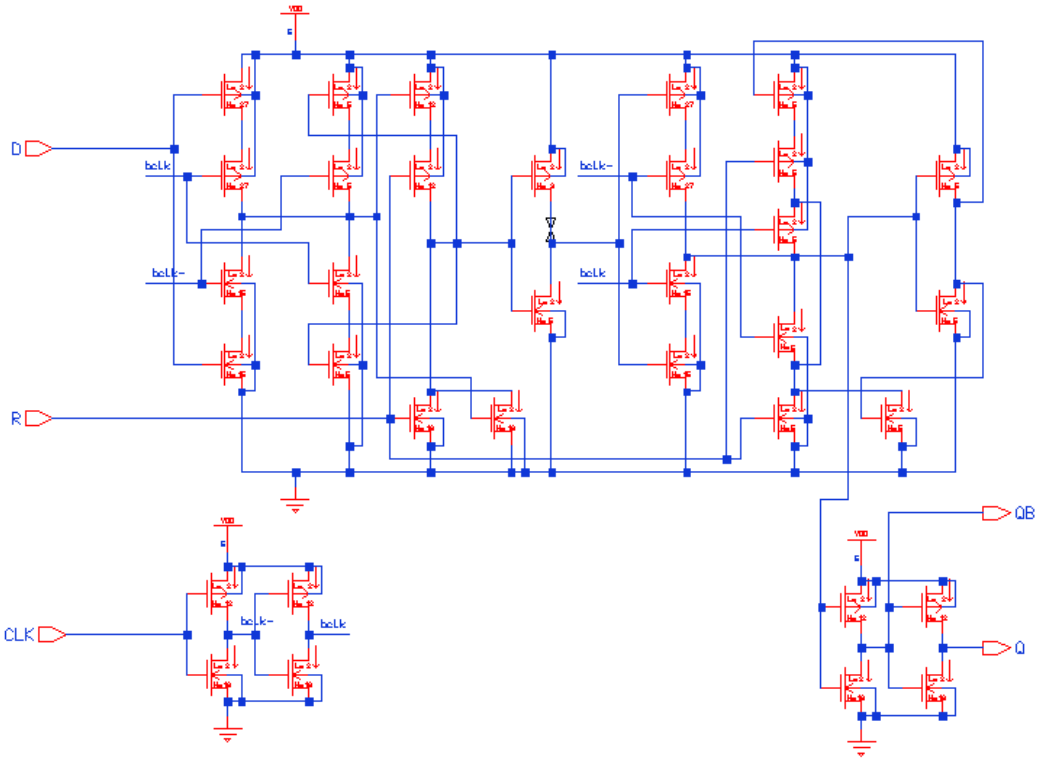


Figure 4.3 D flip-flop with buffer inputs and outputs

For full-custom design methodology used on UMSI chip, uniform height of standard cell is not necessary so that much of chip area was cut down. Generally, most basic gates, such as inverter, NAND gate, NOR gate, etc, are small that 50 lambda is enough. A light change on circuit is performed when a special gate cannot fit into this height. For example, during design of buffers with large ratio of Wp/Wn, parallel-connected transistors are used to replace single large transistor. Another example is that in clock generator, transistors with large ratio of Wn/Wp are substituted by serial-

connected transistors. For multistage cells, such as flip-flop, double height is used so that they also can be placed with other primitives (As shown in Figure 4.4). And different transmission gates are designed to use for digital and analog signal transmitting.

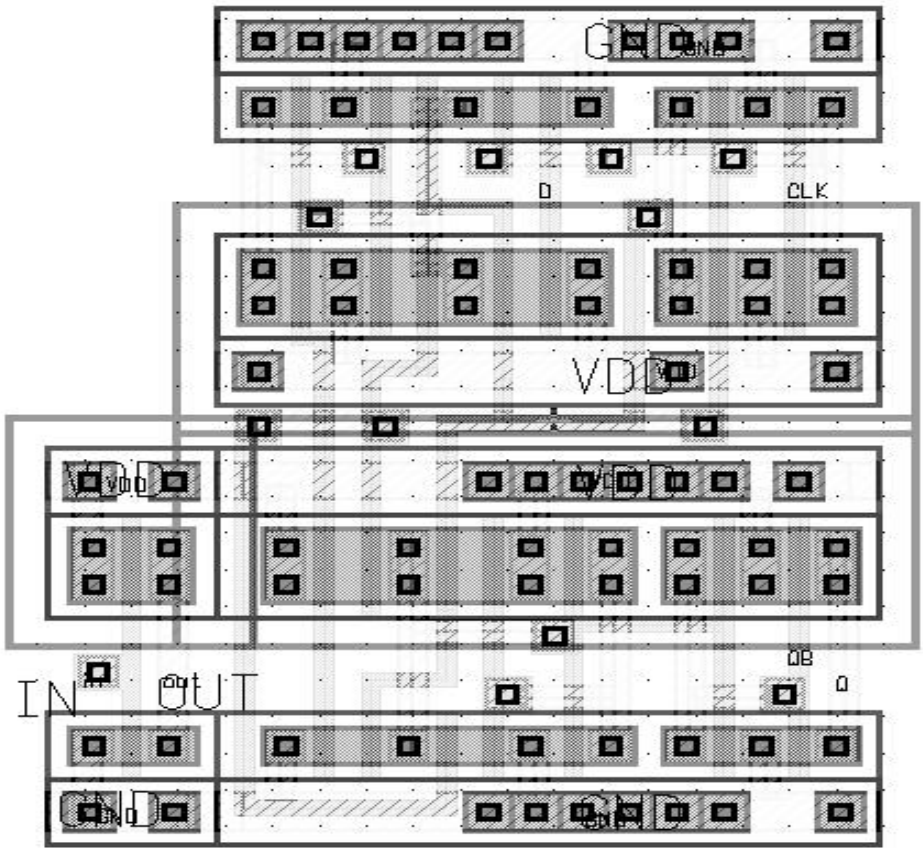


Figure 4.4 Different height cells placed together

4.3 Circuit Building-Block

Full-custom layout techniques should have – at least partially – three features:

1. Minimization of the number of “drawn” devices.

2. “Regularity,” which increases the productivity and decreases the chance of mistakes.

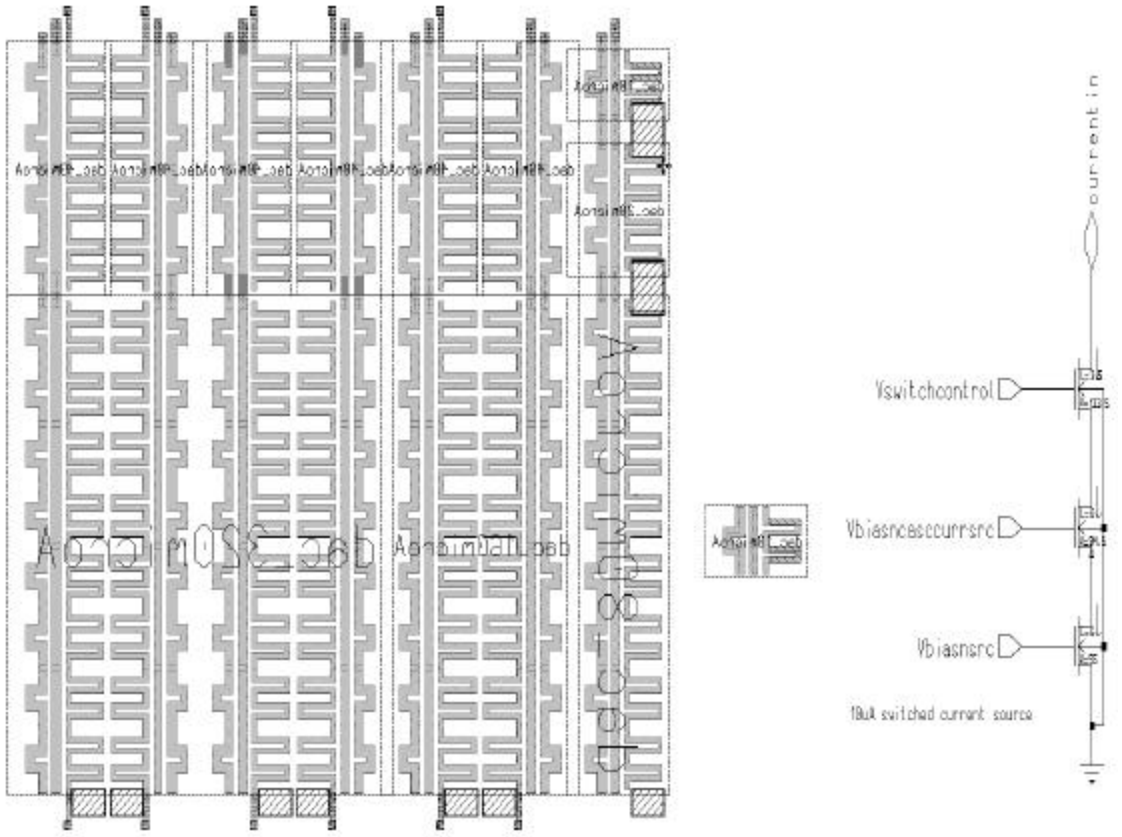
3. Length minimization of conductors in critical paths.

Minimization of drawn devices is accomplished in two steps. First, the logic design must include as many identical cells as possible. This is a *logic design* [9] issue, not a layout issue. Nonetheless, this is the necessary condition to achieve a minimization of drawn devices. The second step consists of *planning* the layout in such a way that all the logically identical cells share the same layout characteristics – that is, aspect ratio, transistor sizes, and location of signal and power/ground terminals. This is a very complex task. To make it possible, sometimes area minimization will sacrifice.

The issue of layout regularity is tightly coupled with design regularity. A regular design aims to use the same logic blocks repeatedly. Same logic blocks should feature the same layout, that is, the same geometrical relationship between inputs, outputs, and power lines. Otherwise, some rearrangement is necessary. 8-bit SIPO shift register appears in many blocks of UMSI chip with other circuit, memory unit and decoder are also repeated blocks, thus, these all can be designed regularly. Therefore, the number of drawn blocks is very small, and more attention can be paid to the placement and routing of irregular blocks and regularization.

Because the speed of UMSI chip is not a critical requirement and so other factors play a more crucial role – design time, costs, and so on – regular methodologies can be fruitfully applied throughout the entire design process, that is, from logic design to layout. The best example is current source of DAC, it is composed of several parallel-connected current sources which is multiple of 10μA current source circuit, shown as Figure 4.5 (b). A regular unit is designed and all other sources are made up by this unit, shown as Figure 4.5 (a). This approach can boost productivity as much as many times over the more conventional “irregular” layout methodologies. The drawback – large

parameters – is not a concern in UMSI chip where short development time, ease of debugging, and yield maximization (i.e., area minimization) are major constraints.



(a) (b)

Figure 4.5 Regularity of DAC current source

The size, geometry, and placement of wells affect the area of bulk CMOS chips significantly, and it is important to optimize the use of wells. Because each well should be tied to the proper potential with substrate contacts, laying out small and sparse wells can waste a significant amount of real-estate. A few, large wells should be laid out. If two wells of the same type are contiguous but not connected together, they should be merged. Therefore, one should avoid alternating substrates of different types. The better method

for standard cell placement is vertically mirroring standard cell row to merge two separate N-wells.

To design digital circuit blocks, an approach called channel routing is use. The concept is using two interconnect layers (Metal1 and Metal2). Cells are placed on the longitudinal sides of the channel. Wires are implemented as a sequence of orthogonal segments using one interconnect layer (Metal1) for the horizontal sections and the other (Metal2) for the vertical ones. This approach has the advantage that it is easily automated. More advanced routers use three or more routing layers, which results in a denser wiring. In general, this means shorter wires, reduced capacitance, and higher performance or lower power. This approach is also used in chip design work with over-the-block routing with Metal3.

Wide devices in analog circuit need careful layout because of gate electrode RC delay. Moreover, the diffusion resistance can slow down the device if few metal contacts are used connect the source to Vdd, and the device is very wide. Different layout techniques can be used depending on the transistor width. Wide transistors can be laid out in parallel-connected transistors. Both source and drain diffusion resistances are decreased drastically by using metal and several diffusion-metal contacts. The source resistance of n-channel (or p-channel) transistors should be kept as low as possible, otherwise significant voltage drop takes place. The voltage drop is proportional to the current that the device can sink (source), and increases as the channel width increases. An insufficient number of source contacts increase the source to Gnd (Vdd) resistance, which worsens the noise margin and creates body effects that slow down the device. Moreover, this also increases the likelihood of latch-up.

4.4 AMI0.5 Pad-Frame

AMI0.5 pad frame is used in UMSI chip design. Basic structure is shown in Figure 4.6. The height of each pad is 300µm, the width is 90 µm.

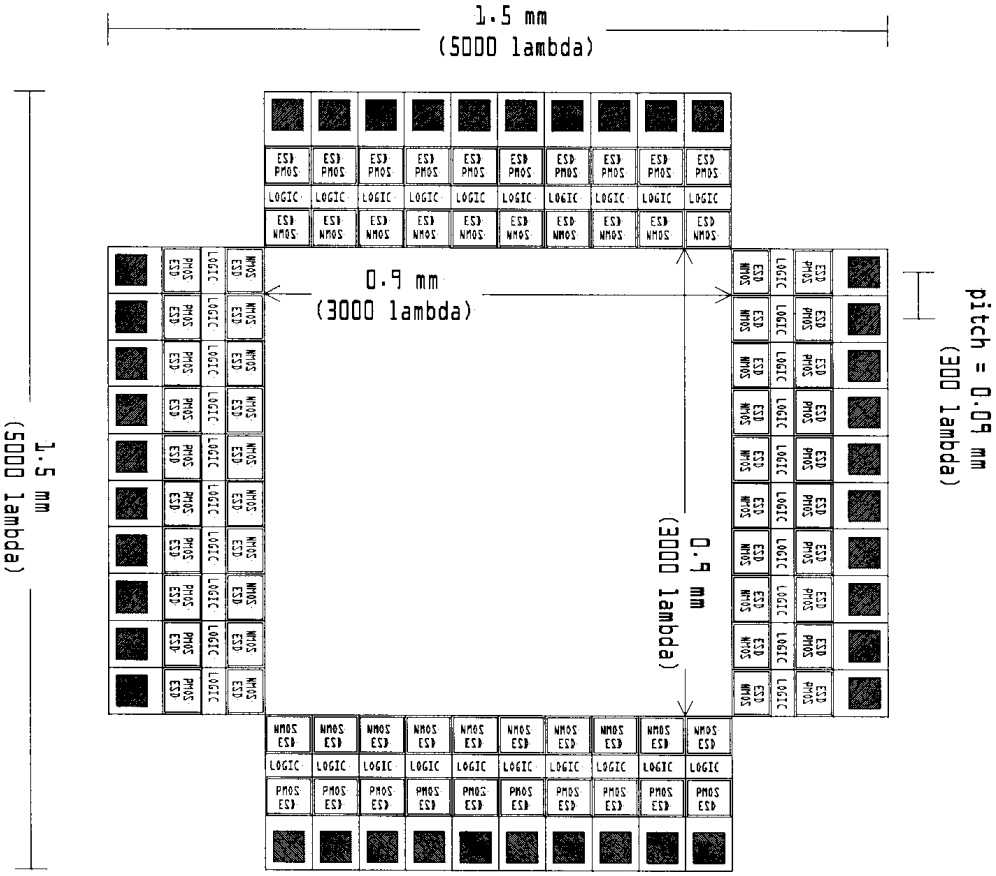


Figure 4.6 AMI 0.5 Hi-ESD Minimum Pad Frame

PadVdd (power pad), PadGnd (ground pad), PadInC (digital input pad), PadOut (digital output pad), and PadIO (analog I/O pad) are used in UMSI pad frame. The area of the final chip is about 2.22µmX2.22µm. Totally 57 pins are used.

4.5 Layout for Testing

Making sure a delivered chip is operating correctly under all possible input conditions is not as simple as it would seem at a first glance. When analyzing the circuit

behavior during the design phase, all the nodes in the network can be accessed unlimitedly. Input patterns can be applied freely and the resulting response at any node also can be observed desirably. But once the chip is manufactured, the only access one has to the circuit is through the input-output pins. For a complex system such as UMSI chip, it is a very lengthy process to bring it into a particular state and to observe the resulting circuit response through the limited bandwidth offered by the input-output pads – if at all possible. It is therefore advisable to consider the testing early in the design process. Some small modifications in a circuit can help make it easier to validate the absence of faults. This approach to design has been called design-for-testability (DFT). DFT is an integral and important part of the design process and should be considered thoroughly. In UMSI design process, two methods for testing are used: scan-based test, and additional circuit test.

Scan-Based Test

Scan-based test is shown in Figure 4.7. Four pins of the UMSI chip provide the testing ports: tclk, tload, tdin, tdout. tclk is the clock input pin for the test port. tload offers the raising edge signal to latch the signals under test. tdin is the data shift in pin. Tout is the data shift out pin.

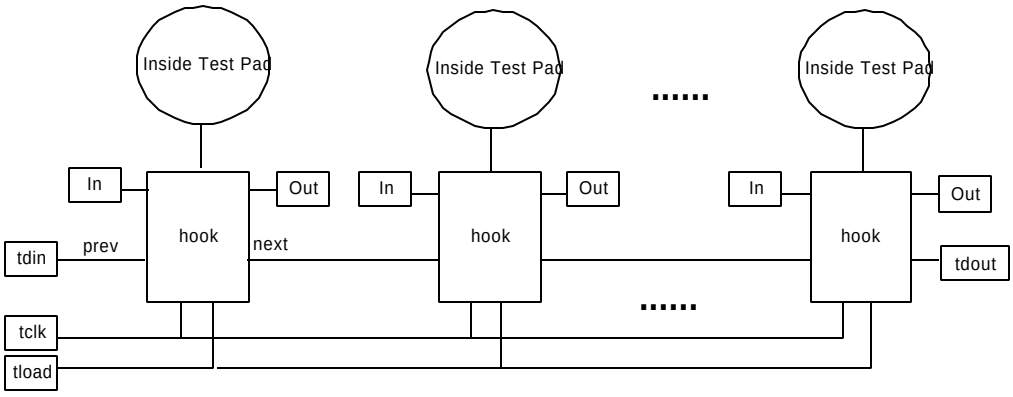


Figure 4.7: Testing chain

When tclk=0, “Out” links directly to “In”. (In --> Out)

When tclk=1 and “next” (or Q) =1, Test Pad links to “Out”. (Tstpad--> Out)

When tload: 0-->1, “Out” data will be latched and saved in D flip-flop. So if we add tclk, the data can be shifted out; if we want to set data of D flip-flop, we can input data from tdin.

When tload=1, Test pad links to “In”. (In--> Tstpad)

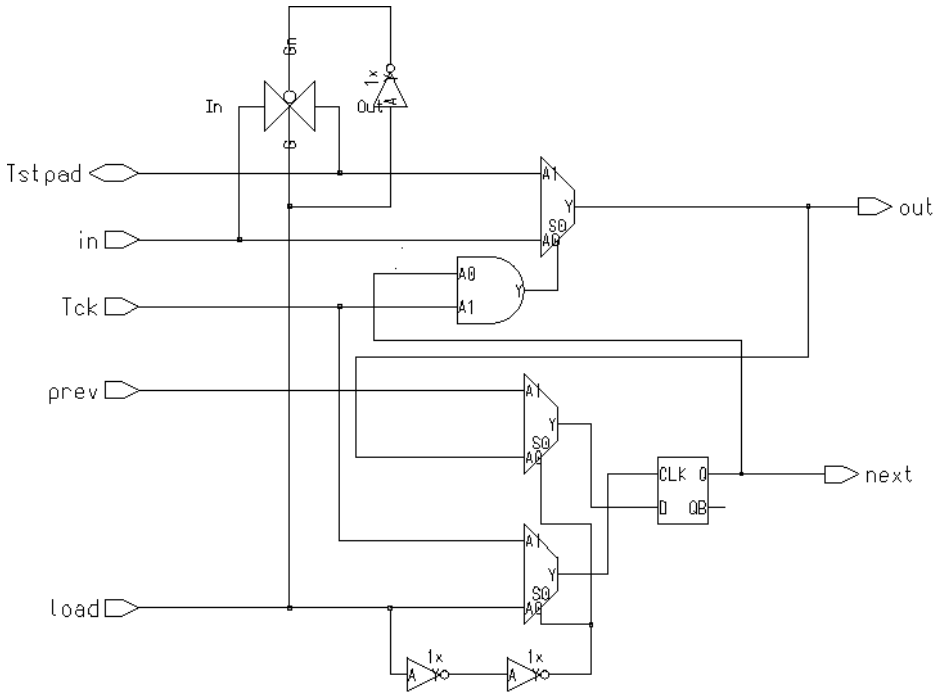


Figure 4.8: The schematic of one testing node (hook)

Applications:

tclk=0 and tload=0: UMSI works in normal mode

tload=1: Test the output of previous module

tload=0, tclk=1 and Q=1: Add test signal for next module from test pad. Q is shifted bit by bit from tdin. So we can enable (Q=1) or disable (Q=0) the test nodes that can add test signal.

tload 0->1 and tclk is added: Read out the "In" status. Sometimes, it is also used as the Q shift-in. The data for shift-in is from tdin that can enable or disable the node function of adding test signal for next module. Figure 4.8 is the schematic of one testing node (hook) on the scan ring.

Additional Circuit Test

To test correctness of basic standard cell, some additional circuits are fed into the chip. These circuits are very common in UMSI circuit. They include D flip-flop with reset, D flip-flop without reset, latch, inverter chain, capacitive amplifier, resistive amplifier, and voltage amplifier; every circuit has their own testing input-output pads and can be probed using probe station to apply and detect signal. If manufactured chip cannot perform expected functions, these basic circuits are first tested to make sure the problem is not due to basic component. And then further test should be performed to find higher level of fault. Also, the parameters obtained from these cells are helpful to future design of standard cell and knowledge of fabrication process.

Besides testing approaches described above, hundreds of testing pads are connected to crucial nodes of circuit to make manufactured circuit more readable and controllable. The concept is using probes to find approximate positions of faults, and then applying correct signals on these nodes to continue testing of other part of circuit.

4.6 Floor-plan, Place and Route of the Chip

To assemble the complete chip and connect the core of the chip to the input and output pads, one last step called place and route is required. The challenge is to combine modules of different shapes and aspect ratios and to realize the connections between them so that the silicon area is well utilized and the overall chip area is kept within bounds. The irregular shape of the modules can, however, lead to very inefficient implementations and huge areas devoted to wiring. To address this deficiency, it is important to consider the global topology of the circuit in the early phases of the design process – even before the modules are actually designed. This effort, called floor-planning determines the silicon estate attributed to each function, resolves the routing path for the major busses, and provides an initial vision on the supply and clock network. Floor-planning is an essential step in any serious design. At the beginning of the UMSI chip design, it is difficult to make floor-plan. This just like a typical chicken-and-egg problem: to come up with a good floor-plan, one needs to know the approximate size and aspect ratio of the basic cells. To know this, they should be designed first. After designing them, the floor-plan indicated that some of the blocks have the wrong aspect ratio and/or terminal locations and have to be designed and laid out again. Doing the floor-plan with just a vague idea about the geometrical characteristics of the cells can lead to geometrical requirements on some blocks that, when one attempts to lay them out, turn out to be impossible to meet. There is no easy solution to this problem. The only information can be used is existing schematics: Relied on the number of transistors that every block has, and the interconnections between them, re-draw the symbols of them, and then place these symbols properly in order to make rough floor-plan. Initially, the largest block, bus interface circuit, is designed first. Smaller blocks, such as analog

readout circuit, are then made based on terminal positions and shape of bus interface circuit.

(a) (b)

Figure 4.9 Pitch matching

Based on the information of the first version of UMSI chip and its blocks, it is possible to produce a floor-plan which is very close to that of the final product during the design of second chip. A feasibility study can then assess whether the circuitry will fit in a single chip, which paths are the longest, etc.

Another important concept in correct layout techniques is pitch-matching. For example, as shown in Figure 4.9, before laying out block A, block B should be laid out; B's width is likely to be wider than that of block A. So it is a worthless effort to minimize the area of block A if the width of block B does not match. All the time spent on this optimization has been wasted. Several kinds of feed-throughs are designed to replenish blanks for pitch-matching and for wire-passing.

Buses and other long interconnections must be laid out in metal. Under no circumstances should polysilicon be used for these lines. When the layout of a block is carried out, metal and polysilicon lines are laid out perpendicularly, because this makes the layout easier, and – when possible – polysilicon lines go through the shortest dimension of the cell. Minimizing the length of polysilicon lines is a critical requirement

in any layout. Common practice is to use polysilicon routing short vertical path, first-metal routing horizontal path, second-metal routing vertical path, and third-metal routing long horizontal path on highest level and power and ground path. Using third metal for power and ground also has the benefit of reducing electromigration effects, because the third metal is closer to the chip surface than the others, and, therefore, heat dissipation can be more effective.

Power distribution is a critical issue in chip layout design. A model for the Vdd and Gnd paths is shown in Figure 4.10

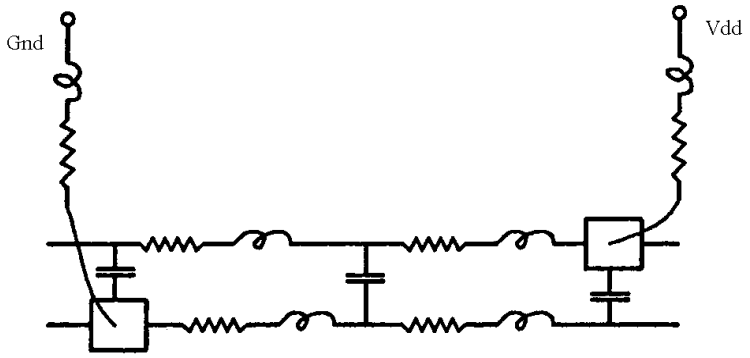


Figure 4.10 Model for Vdd and Gnd paths includes bonding wires

The model includes the bonding wires and metal lines. The bonding wires consist of an inductance and resistance. The metal lines are also modeled with inductance and resistance, and a coupling capacitance connects them together. Besides electromigration effects – current density has to be carefully evaluated – the inductance represents a major danger to proper circuit operation. Voltage drops on power paths (both Vdd and Gnd) not only slow down the circuit because of body effects, but also increase the possibility of latch-up. To avoid these problems and also avoid influence that digital circuit pose on

analog circuit, separate powers are used for bus interface, analog readout, temperature sensor, shock sensor. Those two powers in analog readout are for power management design as discussed before.

Chapter Five

Conclusions

Final chip layout is shown in Figure 5.1. There are 5906 nmos, 5925 pmos, 24 capacitors and 14 resistors in it.

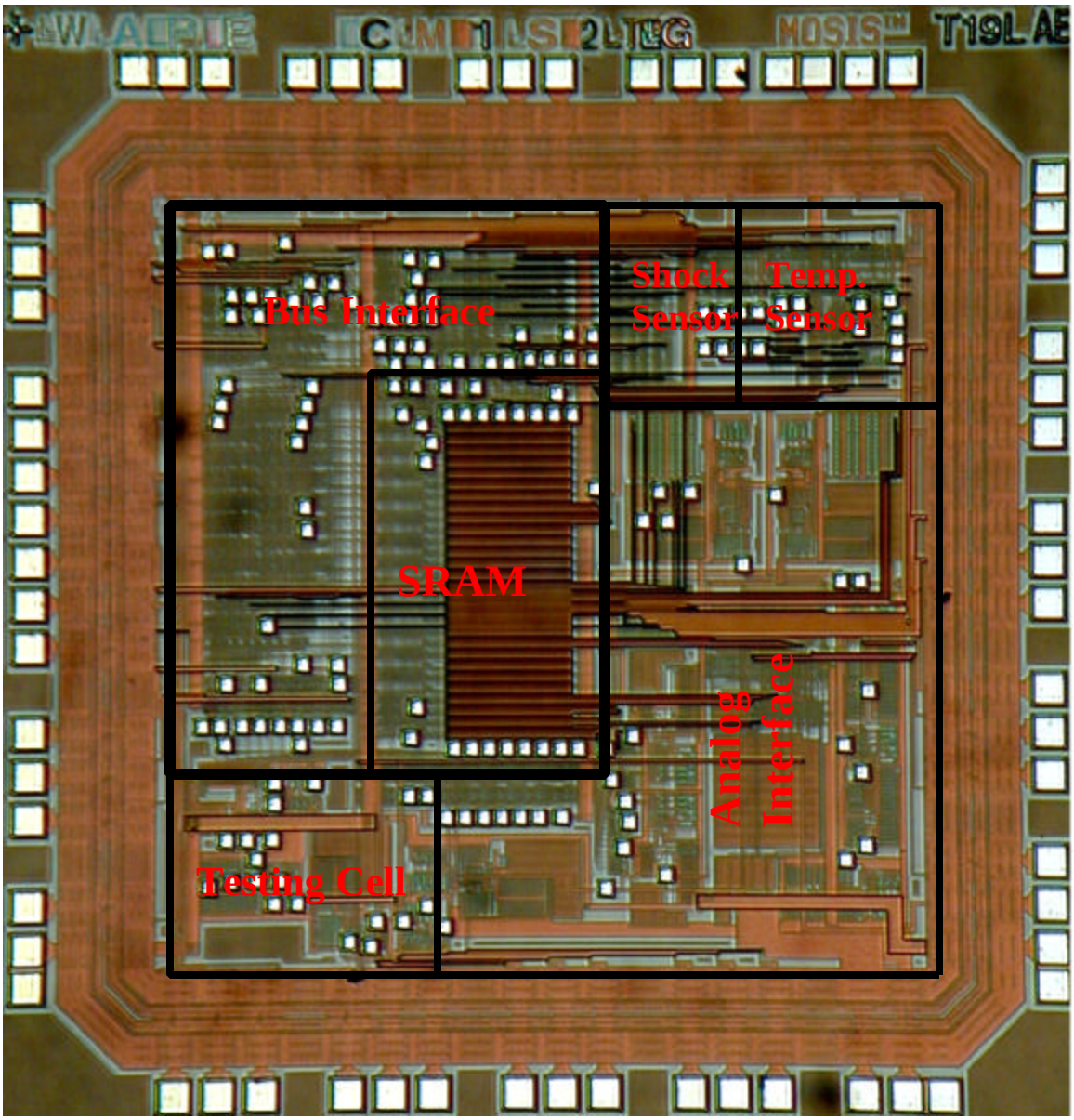


Figure 5.1 Final chip

The following is the list of circuit pins,

- Power pins (5): BVDD for bus interface, AVDD1 and AVDD2 for analog readout circuit, TVDD for temperature sensor, and SVDD for shock sensor.
- Ground pins (2): GND1 for bus interface and shock sensor, GND2 for analog, temperature.
- Sensor bus (8): NTRIG, NIOE, NSDET, DIN, NACK, DOUT, DCLK, NINT
- SPI bus pins (4): SI, SK, STB, SO
- Test pins (4): tclk, tload, tdin, tout
- I/O pins (8): IDI/O (0,7)
- Sensor input pins (8): SenIn (0,7)
- Interrupt pins (8): Shock (0,7)
- D/A pins (1): D/A
- Other pins (2): mode (chip function select), Extcap (testing pin for capacitive readout circuit)
- Clock generator pins (3): CLK, F_1, F_{1_bar},
- Analog circuit testing output (1): V_{out_Analog}
- Analog data output (1): analogout
- Shock sensor input (1): intrexp
- Temperature sensor output (1): SenOut

The fabricated chip has been tested. The signals from chip outputs are correct. Figure 5.2 shows the testing result.

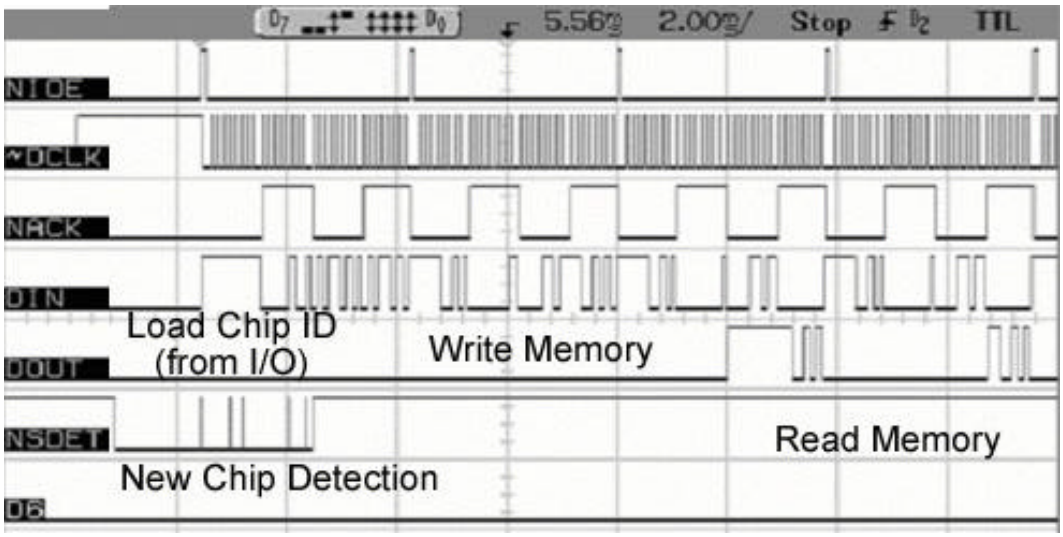


Figure 5.2 Testing Result of Reading and Writing Memory

In this research, an improved sensor communication bus interface circuit has been designed. It is compatible with any microcontroller over a standard sensor bus, and dissipates about 12 mW statically. The circuit has programmable gain and offset control, support sensor self-test, and can connect up to 8 capacitive, resistive, and voltage sensors.

In this chip, the power management is realized by wiring each of separated power pins. For future work, power management circuit should be added into this chip and powering every part of the chip can be controlled by microcontroller. Also, it is noted that the bus protocol as implemented on the UMSI chip could be more closely matched to the IEEE standard. An external EEPROM has been used in this project as TEDS. Although this method simplifies the interface circuit (UMSI) and helps to reduce design cost, in order to maintain consistency with IEEE P1451.2 protocol an internal, on-chip, EEPROM could be implemented. This would provide with uniform address coding better matched to the protocol of the IEEE P1451.2 standard and possibly more generally useful to the sensor industry.

REFERENCES

[1]. S. M. Sze, Semiconductor Sensors, John Wiley & Sons, Inc. 1994.

[2]. A. Mason, "Portable Wireless Multi-Sensor Microsystems for Environmental Monitoring", *Ph.D Thesis*, The University of Michigan, August 2000.

[3]. Z. Wang, "Universal Micro-Sensor Interface Bus", *M.S. Project Report*, The University of Kentucky, August 2001.

[4]. A. Mason, N. Yazdi, A. V. Chavan, K. Najafi, K. D. Wise, "A Generic Multielement Microsystem for Portable Wireless Applications," (Invited) *Proc. IEEE*, vol. 86(8), pp. 1733-1746, August 1998.

[5]. N. Yazdi, A. Mason, K. Najafi, and K. D. Wise, "A Low-Power Generic Interface Circuit for Capacitive Sensors," *Digest, Solid-State Sensor and Actuator Workshop*, Hilton Head Island SC, pp. 215-218, June 1996.

[6]. N. Yazdi, A. Mason, K. Najafi, and K. D. Wise, "A generic interface chip for capacitive sensors in low-power multi-parameter Microsystems," *Elesevier, Sensors and Actuators* vol. 84 (2000), pp. 351-361, December 1999.

[7]. A. V. Chavan, A. Mason, U. Kang, and K. D. Wise, "A Programmable Mixed-Voltage Sensor Readout Circuit and Bus Interface with Built-In Self-test," *Digest, Int. Solid State Circ. Conf.*, San Francisco CA, pp. 136-137, February 1999.

[8]. IEEE Instrumentation and Measurement Society, "IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats," *IEEE Std 1451.2,* 1997.

[9]. Morco Annaratone, Digital CMOS Circuit Design, Kluwer Academic Publishers.

[10]. S. Sunter, "Designing A CMOS Standard Cell Library," *proc. IEEE Custom Integrated Circuits Conference*, May, 1987, pp. 326-329.

[11]. Jan M. Rabaey, Digital Integrated Circuits, Prentice-Hall International, Inc. 1999.

Vita

Kun Zhang was born on June 15, 1975 in Ningxia Province, P.R.China. He received the BS degree in Electrical and Computer Engineering from the Peking University, P.R.China in 1997, and the MS degree in Electrical Engineering from the Chinese Academy of Sciences, P.R.China in 2000. He is currently a Visiting Scholar in the Advanced Micro-System and Circuit (AMSaC) lab at the Michigan State University. From 1997 to 2000, he was a Research Assistant in the Microelectronics R&D Center at Chinese Academy of Sciences, P.R.China. From 2000 to 2001, he was also a Research Assistant in Department of Electrical and Computer Engineering at University of Kentucky.